AN ABSTRACT OF THE THESIS OF

Sean J. Penney for the degree of Master of Science in Computer Science presented on November 28, 2017

Title: Explanations and Information Foraging in Real-Time Strategy Games

Abstract Approved:

Margaret M. Burnett

Assessing and understanding intelligent agents can be a difficult task for users who may lack an artificial intelligence (AI) background. A relatively new area, called "explainable AI," is emerging to help address this problem, but little is known about how to present and structure information that an explanation system might offer. To inform the development of explainable AI systems, we analyzed the "supply" of explanations that experts provided in the real-time strategy domain and conducted an information foraging theory based study to determine if these explanations meet the "demand" of experienced users. Our results showed some consistency between explanations experts offer and what system users demand. We also found foraging problems, however, that caused participants to entirely miss important events and reluctantly ignore other actions, resulting in high cognitive, navigation, and information costs to access the information they needed.

©Copyright by Sean J. Penney

November 28, 2017

All Rights Reserved

Explanations and Information Foraging in Real-Time Strategy Games

by

Sean J. Penney

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented November 28, 2017
Commencement June 2018

Master of Science thesis of Sean J. Penney presented on November 28, 2017.

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Sean J. Penney, Author

ACKNOWLEDGEMENTS

I have been blessed to have a close-knit family who gave me their love and supported my endeavors. My Dad has been my mentor, sharing his enthusiasm for engineering and motivating me to pursue my education. My Mom gave me the unique care only a mother can, encouraging me to achieve my dreams. My brother, Drew, is a truly special friend and, for my real-time strategy research, an on-demand StarCraft II guru.

I have been especially fortunate to study software engineering and human-computer interaction under the guidance of Professor Margaret Burnett. Besides being a world-renowned expert and highly acclaimed professor, Dr. Burnett has been dependably at my side, sharing her research skills and spirited pursuit of new ideas. She is the best advisor I could have ever imagined. Thank you.

I appreciate the opportunity to work with so many great research colleagues. David Piorkowski guided me when I first joined our group working on information foraging theory research. More recently, Jonathan Dodge, Andrew Anderson, and Claudia Hilderbrand have been an amazing team for our explainable artificial intelligence work. Some of the work in this thesis is also presented in our papers being submitted for publication and I thank all my co-authors.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

Table Page

LIST OF APPENDIX FIGURES

Explanations and Information Foraging in Real-Time Strategy Games

Chapter 1 - Introduction

1.1 Real-time strategy games

Real-time strategy (RTS) games are a popular test bed for artificial intelligence (AI) research; platforms supporting such research continue to improve (e.g. (Vinyals, et al., 2017)). Perhaps this is because the RTS domain is challenging for AI due to real-time adversarial planning requirements within sequential, dynamic, and partially observable environments (Ontanon, et al., 2013). These demands are also reflected in real-world environments, so improvements in RTS agents can be applied to other domains, for example, mission planning and execution for AI systems trained to control a fleet of unmanned aerial vehicles (UAVs) in simulated environments.

1.2 Explanations

A human flight specialist may need to evaluate an AI system, but they may not fully understand how the AI system works. Although a user can already observe the system without explanations, machine learning systems typically appear as “black boxes” to end users, as users are not shown why the system behaves the way it does (Kulesza, et al., 2015). Ideally, the system could provide explanations to the human test pilot in a user-friendly way to improve their mental model. If a domain expert making such assessments is not an expert in the complex AI models, there may be a gap between the knowledge they need to make such assessments vs. the knowledge they have in the domain. To close this gap, a growing area known as “explainable AI” aims to enable domain experts to understand complex AI system by requesting explanations. Prior work has shown that such explanations can improve mental models (Kulesza, et al., 2015), (Kulesza, et al., 2010) and users' ability to effectively control the system (Bostandjiev, et al., 2012), (Kulesza, et al., 2012).

To develop a system to provide explanations to domain experts, we must first understand how explanations are structured in the domain and the vocabulary used by expert explainers. We must also understand how a system that provides explanations can support users' information-seeking in the environment.

To this end, we conducted two formative studies. One formative study analyzed the "supply" of expert explanations in the RTS domain. The other formative study investigated the "demand," or how users would consume this information.

First, we looked to "shoutcasters" (sportscasters for "e-sports" such as RTS games) who provide an "expert supply" of explanations. In StarCraft e-sports, two players compete while shoutcasters provide real-time commentary. As communication professionals, they inform an audience they cannot see and from which they cannot receive questions. Despite this, shoutcasters must still be able to determine what information their audience needs to make sense of the game.

Second, we conducted a formative study of experienced StarCraft II players to understand what an RTS domain expert's information needs are -- what they need to have explained, in what sequence, and at what cognitive and time costs. We observed how these players would go about trying to understand and assess an intelligent agent playing the RTS game of StarCraft can help to inform explanation systems in this area.

1.3 StarCraft II

As with other RTS games, players in StarCraft II are involved in three main aspects to destroy their opponent's army. These three aspects consist of economy and production, scouting, and military tactics. Economy and production involve resources to create military buildings and units. Scouting involves sending units to remote areas of the map to see what the opponent is doing. Military tactics involve deploying military units for securing areas of the map, attacking the opponent, or defending one's units and buildings. For readers who may not be familiar with StarCraft II, a detailed game description is included in Appendix 1.

1.4 Information foraging theory

We wanted a higher level of abstraction than features specific to StarCraft for applicability to other RTS environments and connection with other research about humans seeking information. To that end, we turned to information foraging theory (IFT).

IFT has a long history of revealing useful and usable information functionalities in other information-rich domains, especially web environments (e.g., (Pirolli, 2007)) and software development environments (e.g., (Fleming, et al., 2013), (Piorkowski, et al., 2015)). Originally based on classic predator-prey models in the wild, basic IFT constructs are the *predator* (information seekers like our participants) seeking *prey* (information goals) along pathways marked by *cues* (signposts) in an *information environment* (such as the StarCraft replay environment). The predator decides which paths to navigate by weighing the expected cost of navigating the path against the expected value of the location to which it leads.

1.5 Thesis statement

This thesis will study explanation language and information foraging theory in the context of StarCraft II, a real-time strategy game. In particular, we will study game tournament commentary by expert shoutcasters and the information foraging patterns of experienced game players, to investigate the following thesis statement:

> *Commentary by expert explainers can indicate questions these experts answer and how these answers are composed. Formative user studies can reveal information foraging challenges domain experts face. Explanation composition can demonstrate what content an explanation system should offer and information foraging patterns can help us understand how to deliver that content to users of explanation systems.*

Chapter 2 - Literature Review

Mental models, defined as "internal representations that people build based on their experiences in the real world," enable users to predict system behavior (Norman, 1983). Explanations leading to improved mental models of a system help people gain the understanding they need to assess an AI agent. Bostandjiev et al. studied a music recommendation system and found that explanations led to a remarkable increase in user-satisfaction (Bostandjiev, et al., 2012). To improve mental models by increasing transparency of a machine learning system, Kulesza et al. identified principles for explaining (in a "white box" fashion) how machine learning based systems make its predictions more transparent to the user (Kulesza, et al., 2015). In this study, participants used a prototype based on these principles and observed up to 52% improvement in their mental model quality.

Several studies have also found that explanations have been able to improve users' ability to control the system. Stumpf et al. investigated how users responded to explanations of machine learning predictions, finding that participants were willing to provide a wide range of feedback to improve the system (Stumpf, et al., 2007). Kulesza et al. found that the participants who were best able to customize recommendations were the ones who had adjusted their mental models the most in response to explanations about the recommender system (Kulesza, et al., 2012). Further, those same participants found debugging more worthwhile and engaging.

In the domain of intelligent agents in RTS games, there is research into AI approaches (Ontanon, et al., 2013) but there is little research investigating what humans need or want explained. Cheung et al. studied how people watch the RTS genre, creating personas for various types of viewers (Cheung & Huang, 2011). Shoutcasters are one of the personas and Cheung discussed how shoutcasters affect the spectator experience (Cheung & Huang, 2011). Metoyer et al. studied how experienced players provided explanations in the RTS domain to novice users while demonstrating how to play the game (Metoyer, et al., 2010). They developed qualitative coding schemes of the content and structure of the explanations the expert players offered. The work most similar to our own is Kim et al.'s study of intelligent agent assessment in StarCraft (Kim, et al., 2016). Their study invited experienced players to assess skill levels and overall performance of AI bots by playing against them. They observed that the humans' ranking differed from an empirical ranking based on the bots' win rate at AI competitions. Our study differs from theirs in that our participants did not play, but instead strove to understand and explain by interacting with a game replay.

In everyday conversation, people obtain explanations by asking questions. Drawing on this point, Lim et al. categorized questions people ask about AI systems in terms of "intelligibility types" (Lim, et al., 2009b). Their work investigated participants' information demands about context-aware intelligent systems powered by decision trees, determining which explanation types provided the most benefit to users. They found the most often demanded questions were *why* and *why not* (why did or didn't the system do X?). We provide more details of that work and build on it in when we discuss our research question results.

In recognition of the particular importance of these two types of questions, researchers have been working on *why* and *why not* explanations in domains such as database queries (Bhowmick, et al., 2013), robotics (Lomas, et al., 2012), (Rosenthal, et al., 2016), and email classification (Kulesza, et al., 2011). Other research has demonstrated that the intelligibility type(s) the system supported impact which aspects of users' attitudes are affected.

For example, Cotter et al. found that justifying *why* an algorithm works the way it does (but not *how* it works) increased users' confidence (blind faith) in the system --- but did not improve their *trust* (beliefs which inform a full cost-benefit analysis) in the system (Cotter, et al., 2017). Further, it seems that the relative importance of the intelligibility types may vary from one domain to another. For example, Castelli et al.

found that in the smart homes domain, users showed a strong interest in *what* questions, but few other intelligibility types (Castelli, et al., 2017).

We drew on information foraging theory (IFT) to investigate the information that people would seek in the RTS domain. In IFT terms, when deciding where to forage for information, predators (our participants) make cost/benefit estimates, weighing the information value per time cost of staying in the current *patch* (location on the game map or tab with supplemental information) versus navigating to another patch (Pirolli, 2007). Predators, however, are not omniscient: they decide based on their *perceptions* of the cost and value of the available options. Predators form these perceptions using their prior experience with similar patches (Piorkowski, et al., 2015) and the *cues* (signposts in their information environment like links and indicators) that point toward various patches. Of course, predators' perceived values and costs are often inaccurate (Piorkowski, et al., 2016).

IFT constructs have been used to understand human information-seeking behavior in other domains, particularly web navigation (Chi, et al., 2001), (Fu & Pirolli, 2007), debugging (Fleming, et al., 2013), (Kuttal, et al., 2013), (Piorkowski, et al., 2015), and other software development tasks (Niu, et al., 2013), (Piorkowski, et al., 2016), (Ragavan, et al., 2016). To our knowledge, however, IFT has not been used in RTS environments like StarCraft. This work aims to help fill this gap.

Chapter 3 - Research Methods and Procedures

3.1 Research questions

To inform the design of explanation systems, we analyzed shoutcaster commentary and data from our formative study which answered the following research questions:

> RQ1 Explanations: *What relationships and objects do shoutcasters use when building their explanations?*
>
> RQ2 Questions: *What implicit questions do shoutcasters answer and how do they form their answers?*
>
> RQ3 Prey: *What kind of information do domain experts seek, how do they ask about it, and for what reasons?*
>
> RQ4 Foraging Paths: *What paths do domain experts follow in seeking their prey, why, and at what cost?*
>
> RQ5 Decisions and Cues: *What decision points do domain experts consider to be most critical and what cues lead them astray from these decision points?*

RQ1, RQ2, and RQ3 will help to understand what content should be used by explanation systems, while RQ4 and RQ5 help us understand how to present this content to users.

3.2 Shoutcaster study

To study high quality explanations and capable players, we considered only games from professional tournaments denoted as “Premier” by TeamLiquid (http://wiki.teamliquid.net/starcraft2/Premier_Tournaments). Using these criteria, we selected 10 matches available with video on demand from professional StarCraft II tournaments between 2016 and 2017 (Table **1**). Professional matches have multiple games, so we randomly selected one game from each match for analysis. Sixteen distinct shoutcasters appeared across the 10 videos, with two shoutcasters

Shoutcaster team	Tournament	Shoutcasters	Players	Game
1	2017 IEM Katowice	ToD and PiG	Neeb vs Jjakji	2
2	2017 IEM Katowice	Rotterdam and Maynarde	Harstem vs TY	1
3	2017 GSL Season 1 Code S	Artosis and tasteless	Soo vs Dark	2
4	2016 WESG Finals	Tenshi and Zeweig	DeMuslim vs iGXY	1
5	2017 StarLeague S1 Premier	Wolf and Brendan	Innovation vs Dark	1
6	2016 KeSPA Cup	Wolf and Brendan	Maru vs Patience	1
7	2016 IEM Geonggi	Kaelaris and Funka	Byun vs Iasonu	2
8	2016 IEM Shanghai	Rotterdam and Nathanias	ShowTime vs Iasonu	3
9	2016 WCS Global Finals	iNcontroL and Rotterda	Nerchio vs Elazer	2
10	2016 DreamHack Open Leipzig	Rifkin and ZombieGrub	Snute vs ShowTime	3

Table 1. StarCraft II games studied**.**

commentating each time. Here, shoutcaster team (*caster* or *team* for short) differentiates our observed individuals from the population of shoutcasters as a whole.

Shoutcasters should both inform and entertain, so they fill dead air time with jokes. We therefore filtered shoutcasters' utterances by relevance. To do so, two researchers independently coded 32% of statements in the corpus as relevant or irrelevant to explaining the game. We achieved a 95% inter-rater reliability (IRR), as measured by the Jaccard index. (The Jaccard index is the size of the intersection of the codes applied by the researchers divided by the size of the union.) Then, the researchers split up and coded the remainder of the corpus.

To investigate RQ1 (explanation content), we drew content coding rules from Metoyer’s analysis of explaining Wargus games (Metoyer, et al., 2010) and added codes to account for differences in gameplay and study structure. (For ease of presentation, we use the terms "numeric quantity" and "indefinite quantity" instead of their terms "identified discrete" and "indefinite quantity," respectively.) Two researchers independently coded the corpus, one category at a time (e.g., objects, actions, etc..), achieving an average of 78% IRR on more than 20% of the data in each category. One researcher then finished coding the remainder of the corpus.

For RQ2 (implicit questions shoutcasters answered), we coded shoutcasters' utterances by the Lim & Dey (Lim, et al., 2009b) questions they answered. We added a judgment code to capture shoutcaster evaluation on the *quality* of actions. The complete code set will be detailed in the RQ2 results section. Using this code set, two researchers independently coded 34% of the 1024 explanations in the corpus, with 80% inter-rater reliability (Jaccard). After achieving IRR, the researchers split up the remainder of the coding.

3.3 User study

We conducted a pair think-aloud study, where participants worked to understand and explain the behavior of an intelligent agent playing StarCraft II, a popular RTS game (Ontanon, et al., 2013) that has been used for AI research (Vinyals, et al., 2017). We brought participants in as pairs to leverage the social convention of them talking together.

Our setting for the study was StarCraft II replay files. A StarCraft II replay file contains an action history of a game, but no information about the players (i.e., no pictures of players and no voice audio). This anonymized set-up enabled us to tell our participants that one of the players was an AI agent.

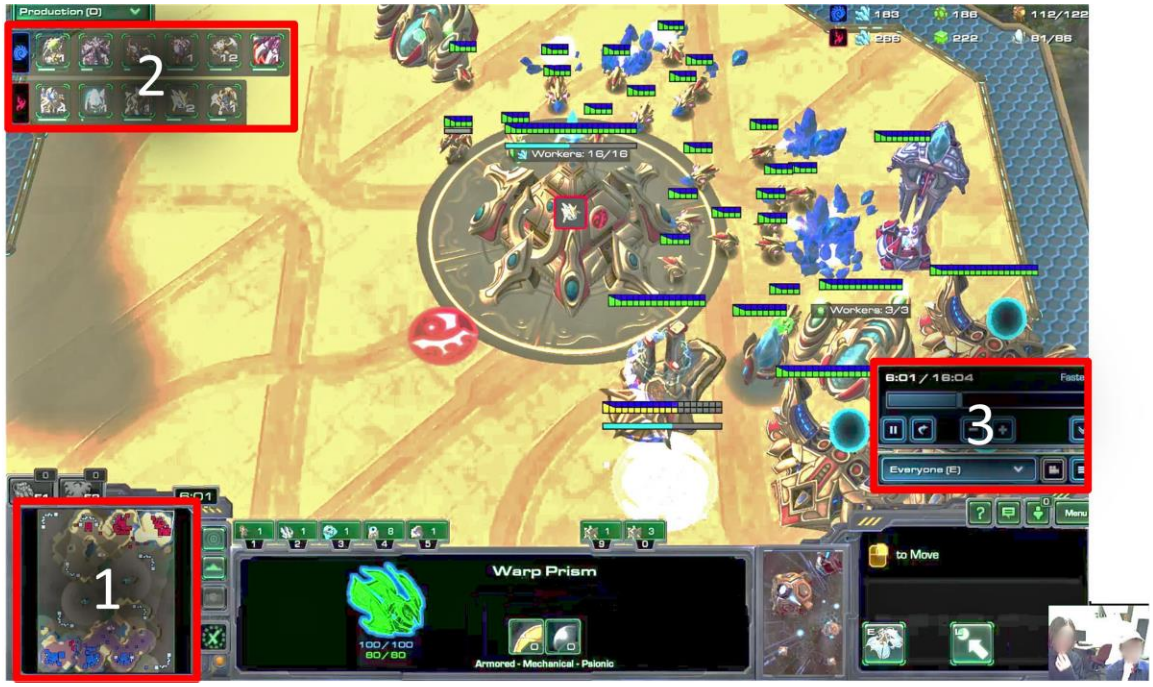

Figure 1. Screenshot from user study. Participants are anonymized (bottom right corner). Important regions marked with red boxes are: 1. The Mini-map offers a birds-eye view enabling participants to navigate around the game map. 2. Participants can use a drop-down menu to display the Production tab for a summary of the build actions currently in progress. 3. Time Controls allow participants to rewind/fast forward, change the speed.

In addition, the participants had functionality to seek additional information about the replay, such as navigating around the game map, drilling down into production information, pausing, rewinding, fast-forwarding, and more, as shown in Figure 1.

The particular match we used was game 3 of the match between professional players *ByuL* and *Stats* during the *IEM Season XI - Gyeonggi* tournament. The IEM tournament series is denoted as a “Premier Tournament,” by TeamLiquid, a multi-regional eSports organization that takes a keen interest in professional StarCraft II. The replay file is public at: http://lotv.spawningtool.com/23979/. The replay we chose to analyze was a representative sample in terms of game flow, e.g., initially building up economy, some scouting, then transitioning to increasing combat (Ontanon, et al., 2013).

We were interested in how participants would go about understanding an intelligent agent's behaviors so we hid the players' names instead displaying them as *Human* and *CPU1*, and told participants that one of the players was under AI control --- even though that was untrue. To encourage them to aim for a real understanding of an agent that might have weaknesses, we also told them the AI was not fully developed and had some flaws. Participants believed our deception and were convinced that the player was an AI. For example, Pair5-P10 speculated about the implementation, "he must have been programmed to spam." Participants did notice, however, the AI at times behaved like a human:

> Pair10-P20: *"Okay, I've not thought of that angle for some reason: The AI trying to act like a human."*

Instead of using deception to simulate an intelligent agent with a human player, an alternative design might use a replay of a game with an intelligent agent playing, but we needed replay files with both interactive replay instrumentation and high-quality gameplay. We were unable to locate an intelligent agent in the RTS domain with high enough quality for our investigation, i.e., without limitations like exploiting "a strategy only successful against AI bots but not humans" (Kim, et al., 2016).

3.3.1 Participants

We wanted participants familiar with the StarCraft user interface and basic game elements, but without knowledge of machine learning or AI concepts, so we recruited StarCraft players at Oregon State University with at least 10 hours of prior experience -- but excluding computer science students. Also, to avoid language difficulties interfering with the think-aloud data, we accepted only participants with English as their primary language. With these criteria, 20 undergraduate students participated (3 females and 17 males), with ages ranging from 19--41, whom we paired based on availability.

Participant	Age	Gender	Major	Casual SC2 hours	Comp SC2 hours	Casual RTS hours	Comp RTS hours
Pair1-P1	41	M	EE	200	100	500	300
Pair1-P2	20	M	ECE	50	20	30	30
Pair2-P3	23	M	CE	10	5	25	55
Pair2-P4	23	M	ME	100	200	50	0
Pair3-P5	21	M	EE	50	0	150	12
Pair3-P6	27	M	CE	15	2	150	10
Pair4-P7	23	M	CE	40	20	20	30
Pair4-P8	28	F	EnvE	200	100	300	30
Pair5-P9	21	M	BE	40	40	100	0
Pair5-P10	19	M	ECE	700	300	50	0
Pair6-P11	22	M	BE	100	2	160	100
Pair6-P12	22	F	EnvE	0	70	0	0
Pair7-P13	22	M	CE	15	60	100	50
Pair7-P14	20	M	BE	35	3	40	0
Pair8-P15	23	M	CE	10	0	100	5
Pair8-P16	22	M	BE	16	1	15	0
Pair9-P17	21	M	PS	90	5	500	80
Pair9-P18	19	M	ME	100	0	20	0
Pair10-P19	24	F	FA	5	5	0	0
Pair10-P20	23	M	EdEn	80	15	50	0

Table 2. Participant demographics. Includes participant casual vs. competitive (Comp.) experience. SC2 is StarCraft II and RTS is any other Real-Time Strategy game.

Participants had an average of 93 hours of casual StarCraft experience and 47 hours of competitive StarCraft experience (Table 2).

3.3.2 Main task procedures

For the main task, each pair of participants interacted with a 16-minute StarCraft II replay while we video-recorded them. The interactive replay instrumentation, shown in Figure 1, allowed participants to actively forage for information within the replay and we gave them a short tutorial of its capabilities. Examples of ways they could forage in this environment were to move around the game map, move forward or backward in time, find out how many units each player possessed, and drill down into specific buildings or units.

Participants watched and foraged together as a pair to try to make sense of the agent's decisions. One participant controlled the keyboard and mouse for the first half of the replay and they switched for the second half. To help them focus on the decisions, we asked them to write down *key decision points*, which we defined for them as, “an event which is critically important to the outcome of the game.” Whenever they encountered what they thought was a key decision point, they were instructed to fill out a form with its time stamp, a note about it, and which player(s) the decision point was about. If participants were confused about what a decision point is, we were prepared to give an example. Participants were all successful, however, in determining what a decision point should be based on our explanation.

3.3.3 Retrospective interview procedures

After the main task, we conducted a multi-stage interview based on the actions the participants took during the main task. To add context to what participants wrote down during the main task, we played parts of our recording of their main task session, pausing along the way to ask why they chose the decision points they did. The wording we used was: “*In what way(s)is this an important decision point in the game?*”

We went through the main task recording again, pausing at their navigations to ask questions drawn from prior IFT research (Piorkowski, et al., 2016). When a participant paused the replay, we asked “What about that point in time made you stop there?” and

"Did you consider stopping on that object at any other point in time?" When a participant navigated to a patch, we asked "What about that part of the game interface/map made you click there?" and "Did you consider clicking anywhere else on the game interface/map?" When a participant navigated away from a patch (or un-paused), we asked "Did you find what you expected to find?" followed by "What did you learn from that click/pause?" and "Did you have a different goal for what to learn next?"

Since there were too many navigations to ask about them all, we sampled pre-determined time intervals to enable covering several instances of each type of navigation for all participant pairs.

3.3.4 Analysis methods

To answer RQ3, we qualitatively coded instances in the main task where participants asked a question out loud, using the code set outlined later in Table 6 in RQ3 results. Our researchers had used this code set on the shoutcaster corpus, in which they achieved sufficient inter-rater reliability (IRR). The same researchers who coded the shoutcaster corpus split up the coding of the user study corpus.

To understand the distinct types of information participants were seeking when asking *what* questions, a group of three researchers first performed affinity diagramming to generate six categories: *Where, Information Availability, Identification, Quantification, When,* and *Resolving Confusion*. Qualitative coding was performed using the following codes based on these six categories. After an IRR of 83% was achieved on more than 20% of the corpus, one researcher coded the remainder of the corpus. The code set will be detailed further in RQ3 results section.

To answer RQ4, we qualitatively analyzed the participants' responses to the retrospective interview questions. To develop a code set to answer the question "*Why was the participant seeking information?"* a group of four researchers started with affinity diagramming to generate groups of answers.

The affinity diagram led to the following codes: *Monitoring State, Updating Game State, Obsolete Domain,* and *New Event*, as shown later in Table 9. Two researchers individually then qualitatively coded the participants' responses using this code set on 20% of the data. Given that our IRR on this portion was 80%, one researcher then completed coding the remainder alone.

To answer RQ5, we qualitatively coded the decision point forms that the participants used during the main task. We developed a code set using affinity diagramming.

The four higher level codes we used were building/producing, scouting, moving, and fighting. We coded 24% of the 228 identified decision points according to this code set and reached IRR of 80%, at which point one researcher coded the remaining data. Coding results are detailed in Table 11 of RQ5 results.

Chapter 4 - Results

4.1 Shoutcaster study results

4.1.1 RQ1 results

> *What relationships and objects do shoutcasters use when building their explanations?*

To inform future explanation systems' content by expert explanations --- the patterns of nouns, verbs, and adjectives/adverbs in these professionally crafted explanations --- we drew upon a code set from prior work (Metoyer, et al., 2010) (see the Methodology chapter). Table 3 shows how much shoutcaster teams used each of these types of content, grouping the objects (nouns) in the first group of columns, then actions (verbs), and then properties (adjectives and adverbs). Frequencies for each code are percentages of the total number of utterances in the corpus but, since some utterances did not have a corresponding code and other utterances fit more than one code, percentages may not add to 100%.

Shoutcasters' explanation sentences tended to be noun-verb constructions, so we began with the nouns. The most frequently described objects were *fighting object, production object,* and *enemy,* with frequencies of 49%, 34%, and 16%, respectively, as shown in Table 3. This is similar to Metoyer's results, where production, fighting, and enemy objects were the three most popular object sub-codes (Metoyer, et al., 2010). As to the actions ("verbs"), shoutcasters mainly discussed *fighting* (40%) and *building* (23%). It is not surprising that shoutcasters frequently discussed *fighting,* since combat skills are important in StarCraft (Kim, et al., 2016) and *producing* is often a prerequisite to *fighting.* This suggests that, in RTS environments, an explanation system may be able to focus on only the most important subset of actions and objects without needing to track and reason about others.

	Objects							Action				Spatial				Temporal				Quantitative			
	Enemy	Fighting	Vision	Production	Environmental	Unspecified	Upgrade	Build/Produce	Fight	Scout	Move	Distance	Point/Region	Size	Arrangement	Order	Timing	Speed	Repetition	Unidentified discrete	Identified discrete	Comparative	Absolute
Pair 1	20%	53%	2%	39%	0%	11%	5%	34%	47%	3%	2%	9%	30%	0%	13%	31%	26%	3%	20%	21%	27%	10%	7%
Pair 2	22%	43%	1%	37%	0%	9%	2%	17%	41%	7%	2%	10%	30%	0%	11%	28%	12%	2%	12%	20%	34%	10%	3%
Pair 3	16%	52%	0%	27%	0%	5%	16%	39%	36%	0%	2%	14%	23%	0%	9%	36%	41%	11%	11%	7%	27%	20%	5%
Pair 4	14%	43%	2%	23%	3%	7%	10%	23%	49%	1%	0%	11%	18%	0%	12%	9%	18%	7%	8%	13%	20%	10%	4%
Pair 5	20%	44%	8%	28%	1%	2%	9%	23%	34%	5%	4%	6%	17%	1%	4%	13%	18%	11%	13%	15%	31%	8%	4%
Pair 6	16%	38%	8%	35%	0%	3%	2%	17%	41%	4%	4%	9%	19%	0%	9%	23%	24%	5%	16%	10%	22%	14%	6%
Pair 7	17%	56%	4%	40%	1%	4%	4%	16%	40%	6%	6%	16%	22%	1%	10%	18%	18%	6%	19%	8%	44%	12%	6%
Pair 8	18%	59%	0%	53%	0%	0%	2%	30%	38%	4%	8%	16%	29%	1%	10%	55%	40%	15%	25%	16%	27%	6%	10%
Pair 9	8%	48%	0%	23%	0%	8%	5%	15%	38%	2%	3%	11%	18%	0%	9%	34%	31%	3%	20%	17%	34%	11%	5%
Pair 10	9%	53%	5%	40%	7%	2%	3%	19%	33%	7%	2%	9%	17%	5%	3%	14%	12%	17%	14%	9%	24%	7%	7%
Mean	16%	49%	3%	34%	1%	5%	6%	23%	40%	4%	3%	11%	22%	1%	9%	26%	24%	8%	16%	14%	29%	11%	6%
Stdev	5%	7%	3%	9%	2%	4%	5%	8%	5%	2%	2%	3%	5%	2%	3%	14%	10%	5%	5%	5%	7%	4%	2%

Table 3. Code occurrence frequencies. Frequencies for each code are percentage of the total number of utterances in the corpus. From left to right: Objects, Action, Spatial, Temporal, and Quantitative codes. Percentages may not add to 100%; see text. Shoutcasters were consistent about kinds of the content they rarely included, but inconsistent about the kinds of content they most favored.

Table 4 shows how shoutcasters' explanations used these concepts *together*, i.e., which properties they paired with which objects and actions.

Shoutcasters were strategic in how they put together these nouns and verbs with properties. Shoutcasters used particular properties with these nouns and verbs to paint the bigger picture of how the game was going for each player and how that tied to the players' strategies. We illustrate, in the next subsections, a few of the ways shoutcasters communicated about player decisions --- succinctly enough for real time.

		Adjectives/Adverbs											
		Distance	Point/Region	Size	Arrangement	Order	Timing	Speed	Repetition	Unidentified discrete	Identified discrete	Comparative	Absolute
Nouns	Enemy	11%	12%	0%	10%	12%	8%	3%	10%	10%	11%	8%	6%
	Fighting object	12%	16%	0%	12%	18%	15%	6%	13%	15%	15%	9%	7%
	Vision object	3%	4%	2%	4%	3%	3%	4%	3%	3%	2%	2%	4%
	Production object	10%	16%	1%	5%	17%	18%	8%	17%	8%	28%	8%	4%
	Environmental object	2%	2%	10%	2%	1%	1%	2%	0%	0%	1%	1%	0%
	Unspecified object	2%	5%	0%	3%	4%	4%	2%	4%	8%	4%	11%	6%
	Upgrade object	1%	1%	0%	0%	6%	10%	11%	3%	1%	10%	6%	4%
Verbs	Build/Produce	3%	7%	0%	2%	16%	21%	12%	18%	7%	20%	6%	3%
	Fight	14%	19%	1%	12%	19%	13%	5%	12%	15%	16%	8%	7%
	Scout	2%	5%	2%	4%	5%	3%	2%	3%	3%	2%	1%	4%
	Move	8%	8%	3%	5%	4%	3%	4%	2%	2%	3%	1%	2%

Table 4. Co-Occurrence Matrix. Across rows: Object (pink) and Action (orange) codes. Across columns: Spatial (green), Temporal (yellow), and Quantitative (blue). Co-occurrence rates were calculated by dividing the intersection of the sub-codes by the union.

“This part of the map is mine!”: Spatial properties

RTS players claim territory in battles with the *arrangement* of their military units, e.g.:

> Shoutcaster Team 3: *“He's actually arcing these roaches out in such a great way so that he's going to block anything that's going to try to come back.”*

As the *arrangement* column in Table 4 shows, the objects that were used most with *arrangement* were *fighting objects* (12%, 72 instances) and *enemy,* (10%, 26 instances). Note that *arrangement* is very similar to *point/region*, but on a smaller scale.

Arrangement of *production object*, such as exactly where buildings are placed in one's base, appeared to be less significant, co-occurring only 5% of the time.

The degree to which an RTS player wishes to be aggressive or passive is often evident in their choice of what *distance* to keep from their opponent and shoutcasters often took this into account in their explanations. One example of this was evaluation of potential new base locations.

> Shoutcaster Team 5: "i*f he takes the one [base] that's closer that's near his natural [base], then it's close to Innovation so he can harass*."

Here, shoutcasters communicated the control of parts of the map by describing *bases* as a *region* and then relating two regions with a *distance*. The magnitude of that distance then informed whether the player could more easily attack. Shoutcasters' utterances that described *distance* along with *production object* referred to the distance between bases or moving to/from a base in 27 out of 44 cases.

"When should I…": Temporal properties

Shoutcasters' explanations often reflected players' priorities for allocating limited resources. One way they did so was using *speed* properties:

> Shoutcaster Team 4: "*We see a really quick third [base] here from XY, like five minutes third."*

Since extra bases provide additional resource gathering capacity, the audience could infer that the player intended to follow an "economic" strategy, as those resources could have otherwise been spent on military units or upgrades. This contrasts with the following example:

> Shoutcaster Team 8: "*He's going for very fast lurker den…"*

The second example indicated the player's intent to follow a different strategy: unlocking stronger units (lurkers). *Speed* co-occurred with *building/producing* most often (12%, 36 instances).

“Do I care how many?”: Quantitative properties

We found it surprising how often shoutcasters described quantities without numbers. In fact, shoutcasters often did not even include *type* information when they described the players' holdings, instead focusing on *comparative* properties (Table 4). For example,

> Shoutcaster Team 1: “*There is too much supply for him to handle. Neeb finalizes the score here after a fantastic game.*”

Here, supply is generic, we do not even know what kind of things Neeb had -- only that he had “too much” of it.

In contrast, when shoutcasters discussed cheap military units, like marines and zerglings, they tended to provide *type* information, but about half of their comments still included no precise numbers. Perhaps if adding one weak military unit that is cheap to build has little impact on army strength, then foraging to get a precise number may not have been worthwhile -- i.e. the value of knowing precise quantities is low. To illustrate, consider the following example, which quantified the army size of both players vaguely, using *indefinite quantity* properties:

> Shoutcaster Team 6: “*That's a lot of marines and marauders and not enough stalkers*”

In the RTS domain, workers are a very important unit. Consistent with this importance, workers are the only unit where shoutcasters were automatically alerted to their death (Figure 2, region 4) and are also available at a glance on the HUD (Figure 2, region 1). Correspondingly, shoutcasters often gave precise quantities of workers (a *production object*). Workers (workers, drones, scvs, and probes) had 46 co-occurrences with *numeric quantities*, but only 12 with *indefinite quantities* (e.g., lot, some, few).

> Shoutcaster Team 2: “*...it really feels like Harstem is doing everything right and [yet] somehow ended up losing 5 workers*”

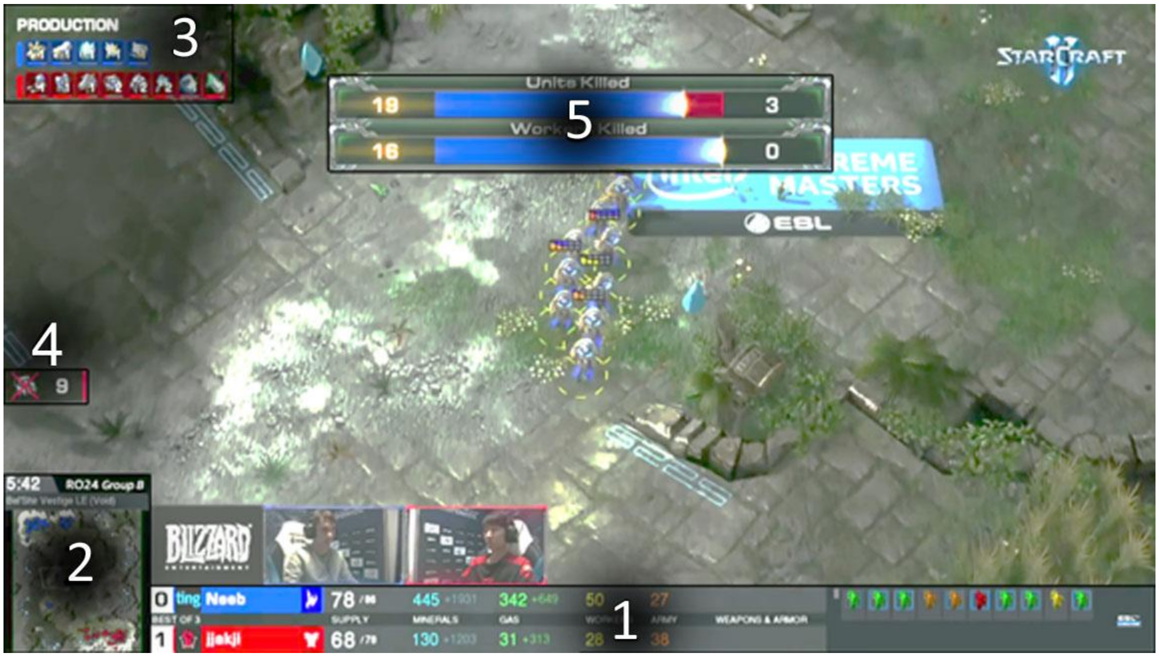

Figure 2. Game screenshot. Screen from an analyzed game shows: 1. HUD (Information about current game state, e.g., resources held, income rate, supply, and upgrade status), 2. Mini-map (Zoomed out version of the main window). 3. Tab (Provides details on demand, currently set on "Production"), 4. Workers killed (Shows that 9 Red workers have died recently), 5. Popup (visualizations that compare player performance, usually shown briefly).

Implications for an interactive explainer

These results have particularly important implications for interactive explanation systems with real-time constraints. Namely, the results suggest that an effective way to communicate about strategies and tactics is to use the critical objects and actions with particular properties that suggest strategies. This not only affords a succinct way to communicate about strategies and tactics, but also a lighter load for both the system and the audience than attempting to build and process a rigorous explanation of strategy.

Specifically, spatial properties can communicate beyond the actual properties of objects to strategies themselves; for example, shoutcasters used distance to point out plans to attack or defend. Temporal properties can be used in explanations of strategies when resource allocation choices determine available strategies.

Finally, an interactive explanation system could use the quantitative property results to help ensure alignment in the level of abstraction used by the human and the system. A player, for example, can abstract a quantity of units into a *single group* or think of them as *individual units*. Knowing the level of abstraction that human players use in different situations can help an interactive explanation system choose the level of abstraction that will meet human expectations. Using properties in these strategic ways may enable an interactive explanation system to meet its real-time constraints while at the same time improving its communicativeness to the audience.

4.1.2 RQ2 results

> *What implicit questions do shoutcasters answer and how do they form their answers?*

Shoutcasters crafted explanations to answer implicit questions (i.e., questions their audience "should be" wondering) about player actions. Thus, drawing from prior work about the nature of questions people ask about intelligent systems, we coded the 1024 shoutcasters' explanations using the Lim & Dey "intelligibility types" (Lim, et al., 2009b). Original work by Lim & Dey investigated information demands from users about intelligent systems powered by decision trees.

Shoutcasters were consistent (Figure 3) in the types of implicit questions they answered.

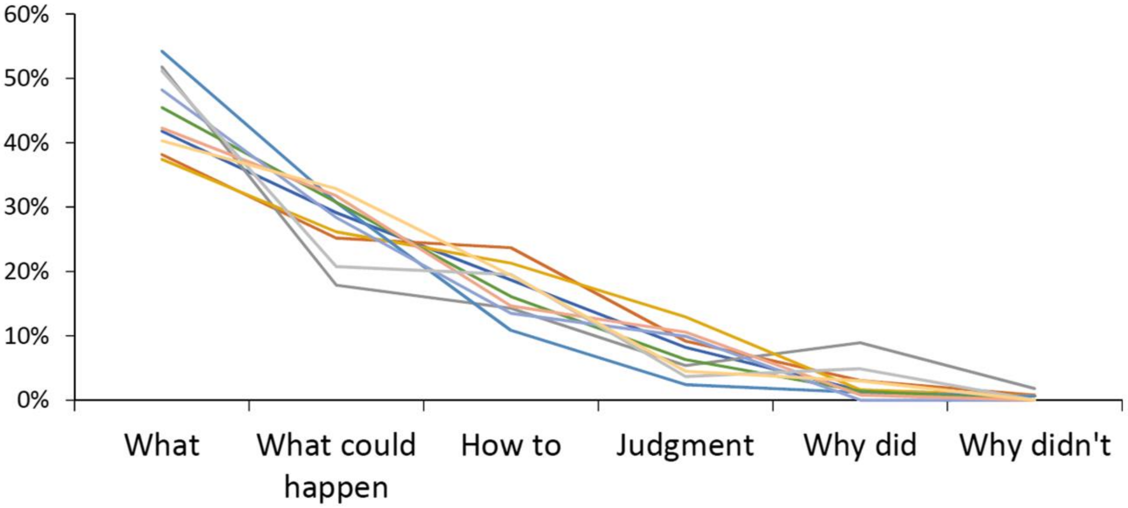

Figure 3. Lim & Dey question frequency. Questions answered by shoutcasters, with one line per shoutcaster team. Y-axis represents percentages of the utterances which answered that category of question (X-Axis). Note how shoutcasters structured answers consistently.

Code	**Frequency**	**Description**	**Example**
What	595	What the player did or anything about game state	“The liberators are moving forward as well”
What could happen	376	What the player could have done or what will happen	“Going to be chasing those medivacs away”
How to	233	Explaining rules, directives, audience tips, high level strategies	“He should definitely try for the counter attack right away”
How good/bad was that action	112	Evaluation of player actions	“Very good snipe there for Neeb”
Why did	27	Why the player performed an action	“...that allowed Dark to hold onto that 4th base, it allowed him to get those ultralisks out”
Why didn’t	6	Why the player did not perform an action	“The probe already left a while ago, so we knew it wasn't going to be a pylon rush”

Table 5. Utterance types. This code set is after the schema proposed by Lim & Dey. We added the code *How good/bad was that action* because shoutcasters judged actions based on their quality.

As shown in Table 5, shoutcasters overwhelmingly chose to answer *what*, with *what could happen* and *how to* high on their list. (The total is greater than 1024 because explanations answered multiple questions and/or fit into multiple categories.)

These results surprised us. Whereas Lim & Dey (Lim & Dey, 2009a) found that *why* was the most demanded explanation type from users, but shoutcasters rarely provided *why* answers.

More specifically, in the Lim & Dey study, approximately 48 of 250 participants, (19%) demanded a *why* explanation. To contrast with our study, only 27 of the shoutcasters’ 1024 utterances (approximately 3%) were *why* answers.

Discussion and implications for an interactive explainer

Why so few *whys*? Should an automated explainer, like our shoutcasters, eschew *why* explanations, in favor of *what*?

One possibility is that shoutcasters delivered exactly what their audience wanted and, thus, shoutcasters' distribution of explanation types was well chosen. This possibility will be explored more in the next results section.

Another possibility is that shoutcasters rarely provided *why* explanations because of the time they required --- both theirs and the audience's. Shoutcasters explained in *real time* as the players performed their actions. It takes time to understand the present, predict the future, and link present to future; and spending time in these ways reduces the time allowable for explaining interesting activities happening in present. This also has implications to the audience's workflow because it takes time for the audience to mentally process shoutcasters' departures from the present, particularly when interesting actions continuously occur.

Even more critical to an explanation system, *why* questions also tend to require extra effort (cognitive or computing resources) because they require connecting two time slices:

> Shoutcaster Team 10: "*After seeing the first phoenix and, of course, the second one confirmed, Snute is going to invest in a couple spore crawlers*."

In this example, shoutcasters had to connect past information (scouting the phoenix, a flying unit) with a prediction of the future (investing in spore crawlers, an air defense structure).

Answering *why didn't* questions was even rarer than answering *why* questions (Table 5). Like *why* questions, *why didn't* questions required shoutcasters to make a connection between previous game state and a potential current or future game state. For example, Shoutcaster Team 2: "*The probe already left a while ago, so we knew it* *wasn't going to be a pylon rush."* The rarity of *why didn't* answers is consistent with the finding that understanding a *why didn't* explanation requires even more mental effort than a *why* explanation (Lim, et al., 2009b).

Shoutcasters found a potentially satisfying approximation of *why*, a combination of *what* and *what could happen***,** the two most frequent explanation types. Their *what* answers explained what the player did , explained what happened in the game, and described the game state. These were all things happening in the present and did not require the additional cognitive steps required to answer *why* or *why didn't*, which may have contributed to its high frequency. Further, the audience needed this kind of "play-by-play" information to stay informed about the game's progression; for example, Shoutcaster Team 4: "*This one hero, marine, is starting to kill the vikings*." When adding on *what could happen*, shoutcasters were pairing *what* with what the player will or could do, i.e., a hypothetical outcome. For example,

> Shoutcaster Team 1: "*...if he gets warning of this he'll be able to get back up behind his wall in*."

Although answering the question *what could happen* required predicting the future, it did not also require shoutcasters to tie together information from *past* and future.

The other two frequent answers, *how good/bad was that action* and *how to*, also sometimes contained "why" information. For *how good/bad was that action*, shoutcasters *judged* an action e.g.:

> Shoutcaster Team 1: "*Nice maneuver from Jjakji, he knows he can't fight Neeb front on right now, he needs to go around the edges."*

For *how to*, shoutcasters gave the audience tips and explained high level strategies. For example, consider this rule-like explanation, which implies the reason "why" the player used a particular army composition: Shoutcaster Team 10: "*Roach ravager in general is really good…"*

The next rule-like *how to* example is an even closer approximation to “why” information. Shoutcaster Team 8: “*Obviously when there are 4 protoss units on the other side of the map, you need to produce more zerglings, which means even fewer drones for Iasonu.*”

In this case, shoutcasters are giving a rule: given a general game state (protoss units on their side of the map) the player should perform an action (produce zerglings). But the example does more; it also implies a *why* answer to the question “Why isn't Iasonu making more drones?” Since this implied answer simply relates the present to a rule or best practice, it was produced at much lower expense than a true *why* answer that required tying past events to the present.

4.2 User study results

4.2.1 RQ3 results

> *What kind of information do domain experts seek, how do they ask about it, and for what reasons?*

To understand how predators seek prey in the RTS domain, we analyzed questions participants asked during the main task. To situate our investigation in the literature of humans trying to understand AI, we coded the utterances using the same Lim & Dey intelligibility types (Lim & Dey, 2009a) (Table 6) that we used to code shoutcaster explanations. Table 7 presents the same information in more detail, broken out by participant pairs. Doing so also allowed us to explore if shoutcasters delivered what users want.

The results again conflicted with other work. Although prior research has reported *why* questions to be much in demand (Lim & Dey, 2009a) (Lim, et al., 2009b), only 10% of our participants' questions fell into the *why* and *why didn’t* categories combined (Table 6). Over 70% of our participants' questions pertained to *what*, as shown in Table 6.

Intelligibility Type	**Frequency**
What: What the player did or anything about game state -Pair3-P5: *"So he just killed a scout right?"*	148 (73%)
What could happen: What the player could have done or what will happen -Pair5-P10: *"What's he gonna do in response?"*	16 (8%)
Why did: Why the player performed an action -Pair10-P20: *"What was the point of that?"*	14 (7%)
How to: Explaining rules, directives, audience tips, high level strategies -Pair3-P5: *"You have to build a cybernetics core, right?"*	9 (4%)
How good/bad was that action: Evaluation of player actions -Pair10-P19: *"Like, cleary it didn't work the first time, is it worth it to waste four units the second time?"*	8 (4%)
Why didn't: Why the player did not perform an action -Pair10-P20: *"why aren't they attacking the base?"*	7 (3%)

Table 6. Intelligibility types. The code set is slightly modified from the schema proposed by Lim & Dey. We added *How good/bad was that action* because the users wanted an evaluation of agent actions.

Question	**Total**	**Pair 1**	**Pair 2**	**Pair 3**	**Pair 4**	**Pair 5**	**Pair 6**	**Pair 7**	**Pair 8**	**Pair 9**	**Pair 10**
What	148	2	3	41	1	6	14	10	1	8	62
What could happen	16		1	1		3	1	1		2	7
Why did	14			2			3	1			8
How to	9	1		3							5
How good/bad was that action	8			3		3					2
Why didn't	7	1		1							5
Total	202	4	4	51	1	12	18	12	1	10	89

Table 7. Lim & Dey questions between participants. Questions participants asked each other, by participant pair. Note how often *What* questions were asked, both by the population of participants and by a few individual pairs, where it was particularly prevalent.

Types of *what* questions	**Frequency**
Identification of noun/verb: Question about what an object in the current game state game is, or what action is taking place - Pair9-P17: “Is than an Overseer there?”	59 (43%)
Quantification: Question about quantity of object in the current game state - Pair10-P19: “Wait, second Gateway or first Gateway?”	25 (18%)
Temporal: Question about prior game state - Pair6-P12: “When did he get zerglings?”	24 (17%)
Resolving Confusion: Question to clarify current game state - Pair3-P5: “What’s going on over here?”	19 (14%)
Where: Question about location of object in current game state - Pair10-P20: “Where did that probe go?”	7 (5%)
Information Availability: Question about what information about game state is available to player - Pair3-P5: “Aren’t they seeing each other?”	4 (3%)

Table 8. *What* question frequency. Frequencies of the question types participants asked during the main sessions.

In both our user study and in the previous analysis of shoutcasters, *what* was the most popular question asked/answered. Since shoutcasters are hired to provide what game audiences want to know, their consistency with our participants' questions suggest that this distribution of questions was typical for the domain.

The many flavors of “*what*” prey

When asking these *what* questions, what types of prey were participants seeking? We identified six prey categories that participants sought when asking *what* questions, with descriptions and frequencies shown in Table 8.

The most common *what* questions participants asked when pursuing prey were questions to identify an object/action in the current game state (43%), or quantifying an object (18%). These questions sometimes involved “drilling down” to find the desired information, which could be expensive. For example, the following question required the participants to drill down into several structures on the map to answer it:

Pair3-P6: *“Is the human building any new stuff now?”*

Although we did not focus on foraging costs when analyzing intelligibility types, we did observe that navigating in pursuit of this kind of prey, which required “drilling down,” was often costly. The least expensive way was navigating via a drop-down menu (2 clicks) in region 2 of Figure 1, but participants instead often foraged in other ways. For example, to answer their question about “building new stuff,” Pair3-P6 made seven navigations by navigating to several unit producing structures on the map, into a structure, and then on to the next.

Shoutcasters' comments met the participants' interest in these questions: 56% of shoutcasters' *what* comments answered questions about identifying an object/action and 28% of shoutcasters’ *what* comments answered questions about the quantity of an object. As an example of the match to shoutcasters' comments, question about identifying units: Pair6-P12: “*I think, well, we have a varied composition, besides roaches and what are these?*” would be well-matched to shoutcaster explanations such as this:

> Shoutcaster Team 3: “*We have the ravagers now coming up.”*

Questions from participants about the quantity of an object, such as Pair7-P14 asking “*Does he have any zealots or stalkers*?” could be answered by shoutcaster explanation like the following:

> Shoutcaster Team 2: “*Couple of stalkers, I’m not even sure if blink is done yet.”*

This suggests that shoutcasters be used as a possible content model for future explanation systems, given that shoutcasters’ “supply” of explanations met participants’ “demand” for their two most popular types of *what* questions.

Another common prey pattern was asking *what* questions about prior state. Questions about past states and when they occurred comprised 17% of their *whats*.

> Pair3-P6: “*When did he start building [a] robotics facility?*”

Shoutcasters sometimes gave answers to temporal questions, although at a much lower rate (3%) than our participants asked them. For example:

> Shoutcaster Team 2: “*The probe already left a while ago…*”

This discrepancy likely occurred because shoutcasters had to provide commentary in “real time,” and could not go back in time to get specific timestamps like our participants did. Participants sometimes went back in time to fill in temporal knowledge gaps because they needed timestamps for the decision point forms.

The next most common prey pattern was at a higher level of abstraction than the specific units or events, aiming instead toward more general understanding of what was going on in the game. These *what* questions arose in 14% of the instances of *what* questions. For example, Pair10-P20 asked, “*What's going on over there*?” in which “there” referred to a location on the map with military units that could have been gearing up for combat. We did not count the number of shoutcaster comments that answered this question because we could not narrow them down in this way. That is, although many of their comments *could* be said to be applicable to this type of question, the same comments were also applicable to more specific questions.

For example:

> Shoutcaster Team 7: “*This is about to get crazy because [of] this drop coming into the main base [and] the banelings trying to get some connections in the middle.*”
>
> Shoutcaster Team 9: “*I like Elazer's position; he's bringing in other units in from the back as well.*”

Questioning the unexpected

Lim & Dey reported that when a system behaved in unexpected ways, users' demand to know *why* increased (Lim & Dey, 2009a). Consistent with this, when our participants saw what they expected to see, they did not ask *why* or *why didn’t* questions. For example, Shoutcaster Team 4 and Shoutcaster Team 5 did not ask any *why* or *why didn’t* questions at all as shown in Table 7. Instead, they made remarks like the following:

Pair4-P7: “*the Zerg is doing what they normally do.”*

Pair4-P8: “*[The agent is] kind of doing the standard things.”*

Pair5-P10: “*This is a standard build.”*

In cases of the unexpected, however, a *what* prey pattern arose, in which participants questioned the phenomena before them. We counted nine *what* questions of this type:

Pair9-P17: “*...interesting that it's not even using those.”*

Pair10-P19: “*I don't get it, is he expanding?*”

Pair10-P19: “*Wow, what is happening? This is a weird little dance we're doing.*”

Pair10-P20: “*<when tracking military units> What the hell was that?*”

The unexpected also produced *why* questions. About half of the participants' *why* and *why didn’t* questions came from seeing something they had not expected or *not* seeing something they had expected. For example:

Pair1-P1: “*<noticing a large group of units sitting in a corner> Why didn't they send the big army they had?*”

Pair10-P19: “*Oh, look at all these Overlords. Why do you need so many?*”

Implications for a future interactive explanation system

Using the Lim & Dey intelligibility types (What, Why, etc.) to categorize the kinds of prey our participants sought allowed for direct comparisons to how shoutcasters provide commentary. This produced implications for shoutcasters as possible “gold standards” for informing the design of a future automated explanation system in this domain. For example, the high rate of *What* questions from participants matched reasonably well with a high rate of *What* answers from shoutcasters. Drawing explanation system design ideas from these expert explainers may help inform the needed triggers and content of the system's *What* explanations.

The dominance of *what* questions also points to participants' prioritizing of state information in this domain. Identification of noun/verb *whats* were about objects or actions in a state they did not know about, quantification *whats* where about identifying quantities of game objects in order to gain details about the state, temporal *whats* were about past states they either hadn't seen or had forgotten and higher-level *whats* were about understanding the purpose of a current or emerging state. Further, shoutcasters exceeded the participants' rate of w*hats* in the two most popular categories with their explanations. This suggests that, in the RTS domain, an explanation system's most sought-after explanations may be its explanations relating to identification of nouns, verbs, or quantities.

As noted in prior research, unexpected behaviors (or omissions of expected behaviors) led to increases in questions of both the w*hat* and the w*hy* intelligibility types (Lim & Dey, 2009a). If an explanation system can recognize unexpected behavior, it could then better predict when users will want *why* and w*hat* explanations to understand the differences in expected and actual behavior. One way to accomplish this would be to compare agent behavior against standard “build orders” that human players follow and look for deviations.

Finally, the cost of navigating to some of the prey became expensive, which points to the need for explanation systems to keep an eye on the cost to users of obtaining that information. In this section, this came out in the form of navigation actions. The next section will point to costs to human cognition as well.

4.2.2 RQ4 results

> *What paths do domain experts follow in seeking their prey, why, and at what cost?*

Various cognitive costs were incurred by participants by following paths to find prey. As an information environment, RTS games have foraging characteristics that set them apart from other information environments previously studied from an IFT perspective, such as web sites (Pirolli, 2007) and programming IDEs (Piorkowski, et al., 2016). These previously studied domains are relatively static, with most changes occurring over longer periods. In contrast, an RTS information environment changes rapidly and continually, driven by actions that do not originate from the foragers themselves. As we will see, this caused participants to spend some time monitoring the overall game state, waiting for a suitable cue to appear for them to investigate further.

The number of paths a forager might follow in an RTS information environment increases with the complexity of the game state, but path *lengths* tend to be short. This is conceptualized in Figure 4. This means that most questions are answered within a few navigations. In foraging environments like IDEs, however, there might only be a few interesting links from any single information patch, but some can lead to lengthy sequences of navigations (e.g., the “Endless Paths” problem (Piorkowski, et al., 2016)).

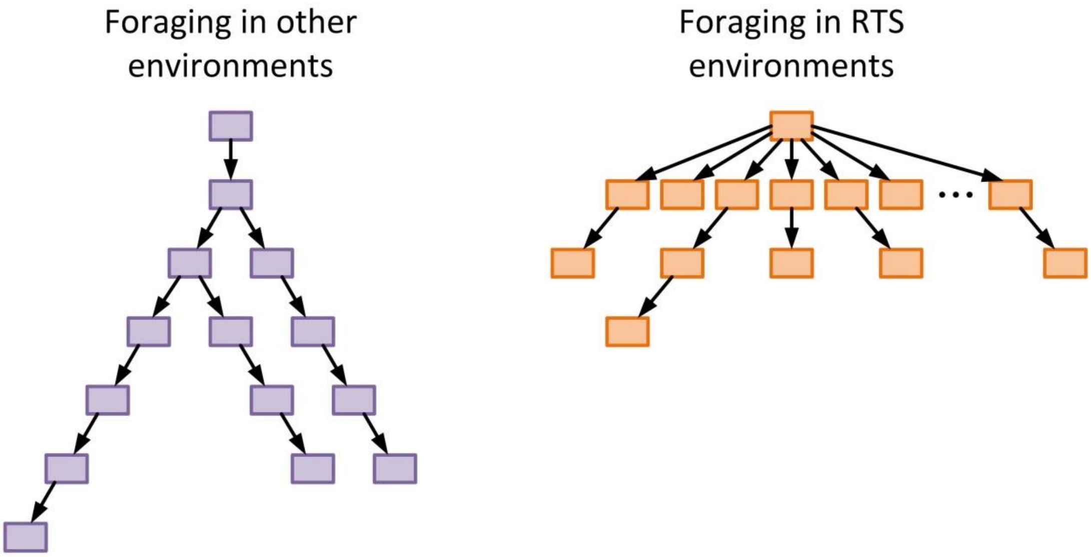

Figure 4. Foraging in prior environments vs. RTS domain. Conceptual drawing to contrast foraging in the RTS domain with previously studied foraging. (Left): Information environments considered by past IFT literature look like this, where the paths the predator considers are few, but sometimes very deep. This figure is inspired by a programmer's foraging situation in an IDE (Piorkowski, et al., 2016). (Right): Foraging in the RTS domain, where most navigation paths are shallow, but with numerous paths to choose from at the top level.

<u>Foraging in the RTS domain</u>

Interestingly, there was hardly any difference between RTS foraging and other environments at first. During the early stages of a game, there are very few units, buildings, or explored regions for users to navigate to, so foraging is relatively straightforward. As one participant put it:

> Pair7-P14: *"There is only so many places to click on at this point."*

As long as this remained the case, each relevant path could potentially be carefully pursued, similarly to an IDE. Four participant pairs (2, 4, 7, 9) paused the replay for an average of 90 seconds within the first 1:30. They studied individual objects and actions with a great deal of scrutiny, which was surprising considering the sparse environment.

Reasons for participants' path choices code set	**Frequency**
Monitoring State: Continuous game state monitoring, such as watching a fight -Pair4-P7: "*I wanted to see how the fight was going.*"	65
New Event: Attending to a new event for which participants wished to satisfy curiosity about -Pair2-P4: "*I saw there was a new building.*"	36
Update Game State: Updating potentially stale game information that the participant explicitly stated prior knowledge about -Pair1-P2: "*I was mainly looking at the army composition, seeing how it had changed from the last fight.*"	29
Obsolete Domain: Explicitly using domain info that may not be current, such as game rules (e.g., what buildings can produce) -Pair3-P5: "*I mainly clicked on the adept because I'm more familiar with [a previous version of the game].*"	11

Table 9. Participant path choice reasons. Reasons path choices code set, with examples and frequency data, to answer the question "Why was the participant seeking that information?"

In contrast, later in the game, when 50 of the same unit existed, they received much less attention than when there was just one.

Choosing among many available paths created cognitive challenges for participants. Participants needed to keep track of an increasing amount of information as the match progressed. Each time a player performed an action, which added information, the participants could forage for this new information. If the participant did so, we coded their navigation as a *New Event*, which accounted for 26% of our interviewed navigations (Table 9). For example:

> Pair10-P19: *"...noticed movement in the Mini-map and that the Zerg troops were mobilizing in some fashion. So I guess I just preemptively clicked..."*

The rate of path creation exacerbated the "many paths" problem. Professional StarCraft players regularly exceed several hundred actions per minute (APM) (Wong, 2014). This meant that players performed rapid actions that changed the game state.

Each of these actions not only potentially created new paths; they potentially updated the existing ones. This caused the knowledge the participants had about paths that had not been recently checked to become stale, which in turn led to a strong prevalence of two behaviors.

Update Game State was very common in our data set, indicating that participants often needed to check on paths that may have been updated (21% of interviewed navigations, Table 9).

> Pair8-P15: "*...there's a big force again. Just checking it out to see if anything has progressed from earlier.*"
>
> Pair1-P2: "*I was mainly just looking at the army composition, seeing how it had changed from the last fight, see if they had made any serious changes...*"

Note that this is slightly different from our second behavior, *Monitoring State*, which is like updating game state, but with a nonspecific goal. *Monitoring State* was the most common reason for interviewed navigations (46% of navigations were for monitoring, Table 9), for example:

> Pair5-P9: "*I noticed like the large mass of units on the map and I wanted to know what the player was doing with them.*"
>
> Pair8-P16: "*I was just kinda checking on things. Sort of due diligence keeping an eye on the different happenings that the AI was doing at the time.*"

Since each event and its corresponding cues were only visible for a limited time, paths not chosen promptly by participants quickly disappeared. Further, paths are numerous and frequently updated. Thus, there is a large risk for paths of inquiry to be forgotten or unnoticed as the game proceeds, as in these examples:

> Pair7-P14: "*Oh my gosh, I didn't even notice he was making an ultra-lisk den.*"

> Pair3-P6: “*I didn't notice they canceled the assimilator.”*

Many rapidly updating paths: coping mechanisms

Our participants responded to this issue in several ways. First, some participants chose a path and stuck to it, ignoring the others. Note that this required paying an information cost because contextual information that may have been very important for future decisions could be discarded in the process. This strategy was exclusively followed by 3 pairs (2,7,8), who barely made any temporal navigations during the study, as described in Table 10. These participants analyzed the replay using not much more time than shoutcasters spend. Achieving this speed of analysis, however, required participants to *ignore* many game events.

For example, when asked about desire to click anywhere else, one participant volunteered:

> Pair10-P19: “*Mmm, if I had multiple, like, different screens yeah. But no, that seemed to be where the action was gonna be.*”

	Task time	Real-time ratio	Rewinds	Time-stamp rewinds	Context notes
Pair 1	20:48	1.3	3	1	Rewatched 1 fight.
Pair 2	20:40	1.3			Extensive pause around 1:00 to evaluate game state.
Pair 3	55:08	3.4	12	9	Rewatched fights and fight setup. Slowed down replay during 1 combat.
Pair 4	32:16	2.0	2		Rewatched opening build sequence and evaluated information available to the agent at a key moment. Many pauses to explain game state.
Pair 5	24:23	1.5	5		Rewatched unit positioning, AI reaction to events, and scouting effectiveness.
Pair 6	31:56	2.0	4	6	Rewatched 2 fights.
Pair 7	29:49	1.9			Made no use of time controls other than pausing to write down decision points.
Pair 8	21:27	1.3			Made no use of time controls other than pausing to write down decision points.
Pair 9	39:17	2.4	2	1	Rewatched 1 fight. Slowed down replay for the entire task.
Pair 10	61:14	3.8	Lots	Some	Rewound extensively, in a nested fashion. Changed replay speed many times.

Table 10. Participant task time. Shows time (33:42±14:18 minutes) and time control usage information. Note that the replay file was just over 16:04, so dividing each pair's time by 16 yields the third column, Real-Time Ratio (2.1±0.89). Some of the times participants rewound the replay were because we requested timestamps for events, shown in the fourth column, Timestamp Rewinds. The last column provides any additional context in which replay and pause controls were used.

In this fashion, participants chose to triage game events based on some priority order. In both of the following examples, the participants navigated away from the conclusion of a fight:

> Pair6-P11: "*I wanted to check on his production that one time because he just lost most of his army and he still had some [enemies] to deal with.*"
>
> Pair3-P5: "*I was trying to see what units they were building, after the fight, see if they were replenishing, or getting ready for another fight.*"

The second method our participants used to manage the complexity of paths was to use the time controls to slow down, stop, or rewind the replay. Although pausing to assess the state was fairly common in all groups, rewind behavior yielded more information.

Pairs 3 and 10 rewound the most often (Table 10) and paid higher *navigation* costs to do so, but they viewed these navigations as worthwhile to providing necessary information:

> Pair6-P11: "*I looped back to the beginning of the final fight ... to see if there was anything significant that we had missed the first time around.*"

The cost of doing so was more than just time, however, because the more paths they monitored, the greater the cognitive load:

> Pair10-P19: "*There's just so much happening all at once; I can't keep track of all of it!*"

<u>Implications for a future interactive explanation system</u>

Assessing an agent required considering a great many paths and then choosing just one, or perhaps a few. Most paths, though, were not particularly *long*. Note that this contrasts with previous literature in software engineering, which is characterized by "miles of methods (Piorkowski, et al., 2016)," such as a long sequence of methods in the stack trace. Thus, rapid evaluation and pruning of paths is critical in the RTS domain, but less so in software engineering, where the options to consider are fewer and time pressure is less. In the IDE case, for example, if a developer is fixing an UI bug, they can potentially ignore database code. Thus, triaging in the IDE setting can be easier. One solution for the RTS domain could be a recommender system to help the user triage which path to follow next.

During assessment, participants often forgot about or otherwise interrupted their paths of inquiry. For example, if a new important path appeared, such as a critical battle, either that path or the current path had to be dropped. In another domain (spreadsheet debugging), participants faced with branching paths with multiple desirable directions became more effective when the environment supported a strategy they call “to-do listing” (Grigoreanu, et al., 2010). To-do listing was supported on its own or in composition of other problem-solving approaches, so it could also act as a strategy enhancer. In the RTS domain, perhaps a similar strategy could enable users to carry on with their current path uninterrupted --- but also keep track of the critical battle to come back to later.

4.2.3 RQ5 results

> *What decision points do domain experts consider to be most critical and what cues lead them astray from these decision points?*

When participants were not heading down the “right” path, what cues did they instead follow toward some other path? Also, what did they consider the “right” cues to follow?

To accomplish this, we started with the human's perspective and the foraging paths that result from it: namely, how participants identified behaviors that were of interest.

We therefore asked participants to write down what they thought were the important game events. We defined the term *key decision points* to our participants as “an event which is critically important to the outcome of the game,” to give participants leeway to apply their own meaning. Since all participant pairs were examining the same replay file, we were then able to compare the decision points the different participants selected.

Code	**Total**	**Pair 1**	**Pair 2**	**Pair 3**	**Pair 4**	**Pair 5**	**Pair 6**	**Pair 7**	**Pair 8**	**Pair 9**	**Pair 10**
Expansion	52	7	7	8	8	-	6	3	-	7	6
Building - Rest	69	6	4	7	15	1	7	11	2	12	4
Building - All	114	13	11	15	21	1	12	12	2	18	9
Fighting - All	98	8	4	11	8	4	10	6	8	11	28
Moving - All	26	3	1	5	1	2	2	2	2	3	5
Scouting - All	23	1	2	1	5	1	1	3	-	3	6
Total	228	34	20	40	45	11	31	39	16	42	65

Table 11. Participant decision points. Summary of decision points identified by participants. Sum may differ from totals shown, since each decision point could have multiple labels. Note how prevalent Expansion was within the Building category.

That is, the cues in the information environment were the same for all the participants --- whether they noticed them or not.

Key decision points fell into four main categories: *building/producing, fighting, moving,* and *scouting*. The participants were in emphatic agreement about the most important types of decision points to pursue. *Fighting* and b*uilding* comprised 85% of the 228 total decision points participants identified (Table 11).

In fact, participants showed remarkable consistency about the importance of the *expansion* subcategory of *building*. Eight of the ten participant pairs identified *expansion* decision points, when a player chooses to build a new resource-producing base (Table 11). Extra resources from expanding allowed a player to gain an economic advantage over their opponent because they could build more units:

> Pair1-P2: "*Of course, if you have a stronger economy you will likely win in the end.*"

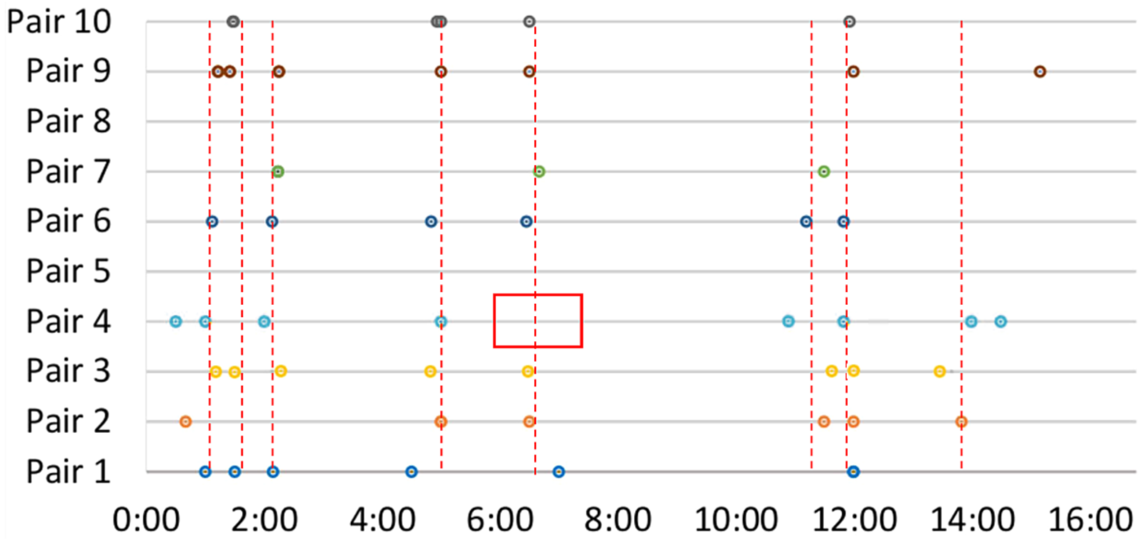

Figure 5. Building/Expansion decision points. Points identified by participants (y-axis), with game time on the x-axis. Expansion events are known to have occurred in the replay file at roughly: 1:00, 1:30, 2:00, 5:00, 6:30, 11:20, 12:00, and 13:45. Each of these times is demarcated on the figure with a red vertical line, often coinciding with decision points. Consider the red box, where Pair 4 failed to notice an event they likely wanted to note, based on their previous and subsequent behavior.

Moreover, because those that identified any *expansion* found at least three, *expansions* seemed to be considered important throughout the duration of the game.

> Pair6-P11: “*... the third base is important for the same reason the first one was because it was just more production and map presence.”*

Even so, they missed some of the cues pointing out expansion decisions. The event logs in the replay file reveal that new bases were constructed at roughly 1:00, 1:30, 2:00, 5:00, 6:30, 11:20, 12:00, and 13:45, each of which is marked with a red line on Figure 5. Only Pair 3 identified decision points for all 8 of these and 7 pairs omitted at least one, with one example highlighted with a red box in Figure 5. Table 11 shows Pair 4 also finding eight *expansion* decision points, but one of those is about the *commitment* to expand, based on building other structures to protect the base, rather than the action of building the base itself.

Since *expansion* decisions were so important to our participants, why did they miss some? “Distractor cues” in the information environment led participants on other paths. Recall that cues are the signposts in the *environment* that the predator observes, such as rabbit tracks. Scent, on the other hand, is what the predators make of cues in their *heads*, such as thinking that rabbit tracks will lead to rabbits. Participants were so distracted by cues that provided an alluring scent, albeit to low-value information, they did not notice the other cues that pointed toward the “expansion” decisions. Distractor cues led participants astray from *expansion* in nine cases and eight of them involved units in combat or potentially entering combat. (The ninth involved being distracted by a scouting unit.) For example, Pair 7 missed the expansion at the 13:45 minute mark, instead choosing to track various groups of army units, which turned out to be unimportant to them:

> Pair7-P14: “*These zerglings are still just chilling.”*

Interestingly, participants had trouble with distractor cues even when the number of events competing for their attention was very low. For example, in the early stages of the game, players were focused on building economies and scouting. There was little to no fighting yet, so it was not the source of distracting cues. We were not surprised that the *Expansion* event at 13:45, when the game state had hundreds of objects and events, was the most often missed (5 instances). We were surprised, however, that even when the game state was fairly simple --- such as at 1:30 when the game had only 13 objects --- participants missed the Expansion events. The extent of distractibility the participants showed even when so little was going on was beyond what we expected.

If decision points went unnoticed in simple game states, what did they notice in complex ones? *Fighting.* All participants agreed *fighting* was key, identifying at least one decision point of that type (Table 11). The ubiquity of *fighting* codes is consistent with Kim et al. (Kim, et al., 2016), who found that combat ratings were the most important to the participant's perception score. *Fighting* provided such a strong scent that it masked most other sources of scent, even those which participants prioritized very highly.

Scouting offers an example of *fighting* leading participants away from other important patches. *Scouting* decision points occurred in the first half of the game, but died out once *fighting* decision points started to occur in the second half of the game. As Figure 6 shows, the start of *fighting* decision points coincides with the time that s*couting* decision points vanish --- even though scouting occurred throughout the game and that participants believed scouting information mattered:

> Pair4-P8: *"But it's important just to know what they're up to and good scouting is critical to know who you are going to fight."*

Implications for a future interactive explanation system

Participants had a tendency to follow cues that were interesting or eye-catching, at the expense of those that were important but more mundane. In this domain, the eye-catching cues were combat-oriented, whereas the mundane cues were scouting oriented. Other domains may have similar phenomena, wherein certain aspects of the agent's behaviors distract from other important cues due to triggering an emotional response in the viewer. As Chi explained, *"…A wealth of information creates a poverty of attention and a need to allocate that attention efficiently among the overabundance of information sources that might consume it* (Chi, et al., 2001)*."* Thus, supporting users' attending to actions that are important but mundane is a design challenge for future interactive explanation systems.

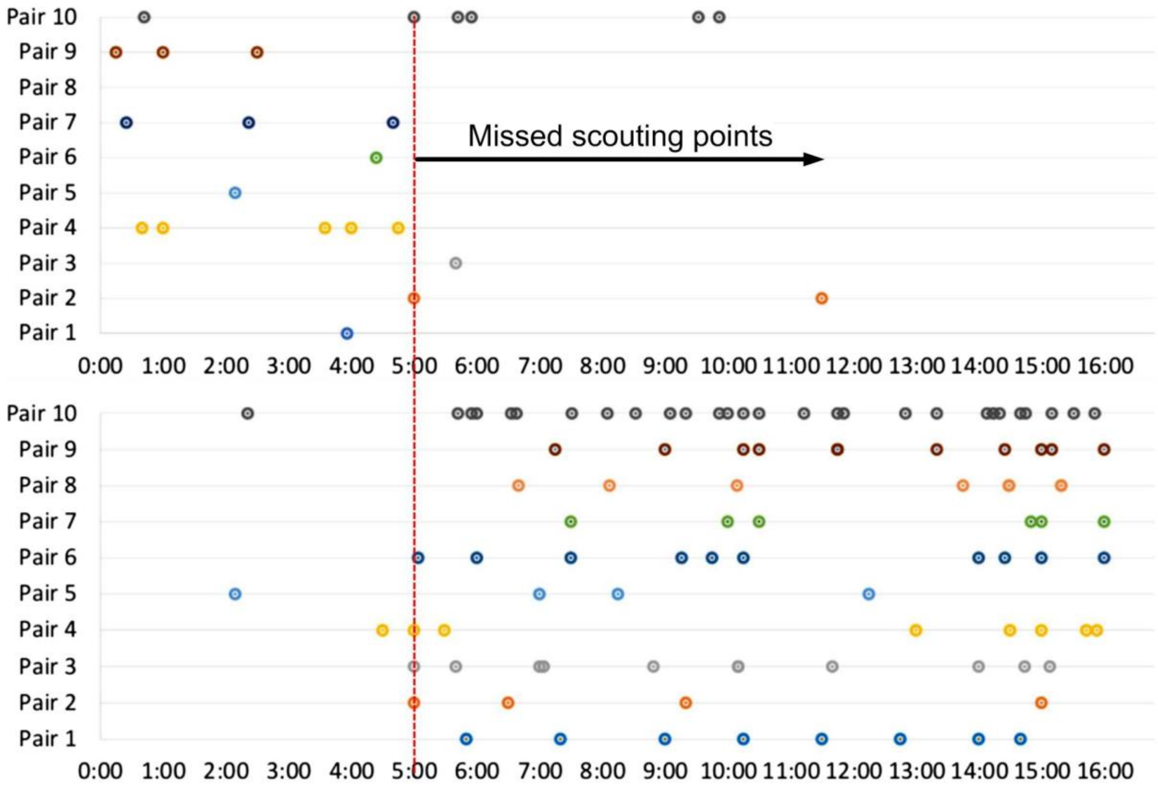

Figure 6. Scouting/Fighting decision points. (Top:) All Scouting decision points identified by our participant pairs (y-axis), with game time on the x-axis. (Bottom:) All Fighting decision points identified, plotted on the same axes. The red line that passes through both images denotes roughly the time at which Fighting events begin. Notice that after this time, many Fighting decision points are identified, but Scouting decision points are no longer noticed often – despite important Scouting actions continuing to occur.

Chapter 5 - Discussion

5.1 Comparison of shoutcaster and user studies

Taken together, Lim & Dey intelligibility types allowed for comparison between the user and shoutcaster studies. Further, both user and shoutcaster studies revealed implications for explanation systems.

As Table 12 shows, w*hat* was the most popular intelligibility type in both the user study and the shoutcaster study. At first glance, many of these *what* explanations may consist of simple play-by-play. Some of these explanations, however, had greater meaning and revealed the strategies and priorities of players. For example, when Shoutcaster Team 4 stated *"We see a really quick third [base] here from XY, like five minutes third,"* shoutcasters were explaining the priorities for allocating resources in addition to giving play-by-play information.

For some types (*what could happen* and *how to*), however, there appears to be a discrepancy. These differences in intelligibility type frequencies can be understood by considering the differences in the roles and objectives of shoutcasters and our participants. Shoutcasters are concerned with how the match plays out to determine the ultimate winner. In this context, *what could happen* is a crucial question, reflecting future events and the likely outcome of the match. The expertise of shoutcasters allowed them to give their audience these predictions of the future. Users, on the other hand, were told to evaluate the intelligent agent so they looked at each move in the context of assessing agent skills. Furthermore, our participants may not have asked *what could happen* questions if they did not expect the question to be answered by their partner or by exploring the interface. If an explanation system could list potential future actions of an intelligent agent, domain experts might utilize this capability. Just as *what could happen* was an important question for shoutcasters, it is also a key question that needs to be answered by explanation systems in order to build user confidence in those systems.

Intelligibility type	**Frequency (shoutcaster study)**	**Frequency (user study)**
What	595 (44%)	148 (73%)
What could happen	376 (28%)	16 (8%)
How to	233 (17%)	9 (4%)
How good/bad was that action	112 (8%)	8 (4%)
Why did	27 (2%)	14 (7%)
Why didn't	6 (<1%)	7 (3%)

Table 12. Lim & Dey intelligibility types: Shoutcasters vs. users. Intelligibility types in the shoutcaster study and user study, ranked by frequencies in the shoutcaster study.

Some *how to* explanations consist of information that domain experts likely do not need (rules about game), so domain experts may not demand *how to* explanations at the rate shoutcasters gave them.

After determining what types of explanations to provide users, the next step at a lower level of detail is to determine the composition of explanations in terms of nouns, verbs, and adjectives/adverbs. We looked at what nouns, verbs, and adjectives/adverbs appeared most frequently, along with how nouns and verbs were typically paired with adjectives/adverbs. This gives insight into how sentences in the explanation system should be composed. Shoutcasters were not, however, always consistent with explanation composition. Production objects were described, for example, at frequencies ranging from 23% to 53% (Table 3). This may have been due to differences in shoutcaster styles, events occurring in the game, or the setup of the game (particular races players chose). For other objects, such as environment objects (frequency of 1% and standard deviation of 2%), it is more clear that explanation systems should not discuss these at all.

After looking at how explanations are formed, we needed to know how domain experts forage for this information to present it to them in an effective way. Information Foraging Theory allowed us to "connect the dots" between our work and other work that has used an IFT perspective. IFT enabled us to abstract beyond game-specific objects like “assimilators” to constructs grounded in a well-established theory for humans' information seeking behaviors. The IFT lens revealed that participants faced difficult foraging problems -- some of which are new to IFT research -- and faced high foraging costs. For example, failure to follow the “right” paths resulted in a high *information cost* being paid, but finding a reasonable path needed to be done quickly due to the ever-changing game environment (at a high *cognitive cost*). Although the user could relax the real-time pressure by pausing the replay, excessive rewinding incurred not only a high *navigation cost* for rewind-positioning and pausing, but also an additional *cognitive cost* of remembering more context.

One open problem in IFT is the “Prey in Pieces” problem. Piorkowski et al. described “Prey in Pieces” as if getting a coffeemaker meant a shopper had to buy individual parts at different stores, then finally piece them together. The cost of going to every store must be paid plus the cost of piecing things together at the end, rather than the cost of going to one store that has a preassembled coffeemaker (Piorkowski, et al., 2016). Our participants encountered this problem when they had to piece together bits of game state information.

Another open problem in IFT is the “Scaling Up” problem (Piorkowski, et al., 2016). This problem was revealed in the domain of IDEs in which foragers (developers) had great difficulty accurately predicting the cost and value of going to patches more than one link away. The problem that the developers faced was a *depth* problem (recall Figure 4). In contrast, in our domain, participants faced a *breadth* “Scaling Up” foraging problem: constantly having to choose which of many paths to follow. The “Scaling Up” problem as a depth problem is still open; so too is the breadth version of it identified here.

5.2 Threats to validity

Every study has threats to validity (Wohlin, et al., 2012). Aspects of our formative study may have influenced our participants to ask less questions in general, such as not asking a question of their partner if they did not expect their partner to be able to answer it. Also, participants took different amounts of time to do the task, ranging from 20 minutes to an hour. Thus, certain participant pairs talked more than others in the main task, creating a form of sampling bias. Furthermore, there were aspects of our study design such as filling out the decision points forms that may result in our study not being completely representative of how domain experts would forage for information to assess an intelligent agent. Threats like these can be addressed only by additional empirical studies across a spectrum of study designs, types of intelligent interfaces, and intelligent agents.

Chapter 6 - Conclusion

In this thesis, we investigated how human experts – RTS shoutcasters – explain strategies in this domain and how domain experts forage for information to assess an intelligent agent in an RTS environment. Utterances of shoutcaster explanations were analyzed to understand questions they *answered* and how they composed their explanations. User studies were conducted with experienced players of StarCraft II, a popular RTS game, in which we observed questions they *asked* and how they prioritized information they sought.

Our results were:

RQ1 *The Explanations*: The composition of shoutcasters' explanations revealed patterns of how they cleverly paired properties ("adjectives and adverbs") with different objects ("nouns") and actions ("verbs") to communicate sophisticated information clearly and concisely. Interactive explanation systems may be able to leverage these patterns to communicate succinctly about an agent's tactics and strategies.

RQ2 *The Questions:* As expert explainers, shoutcasters gave explanations that were feasible to produce and to consume given the time and resource constraints in real-time strategy games.

RQ3 *The Prey*: Participants favored *what* information over the *whys* reported by most previous research and their *whats* were nuanced, complex, and sometimes expensive.

RQ4 *The Paths*: The dynamically changing RTS environment and the breadth-oriented structure of its information paths caused unique information foraging problems in deciding which paths to traverse. These problems led not only to navigation costs, but also to information and cognitive costs.

RQ5 *The Decisions and the Cues*: These costs rendered it infeasible for participants to investigate all of the decision points they wanted. This problem was exacerbated by "distractor cues," which drew participants' attention elsewhere with interesting cues (like signs of fighting), at the expense of information that was often important to participants (like scouting or expansion).

Our results suggest that the information types users demand is consistent with the information types shoutcasters provide. Perhaps most importantly, our results point to the benefits of investigating humans' understanding of intelligent agents through the lens of Information Foraging Theory. For example, the IFT lens enabled us to abstract beyond StarCraft, to reveal phenomena -- such as the frequent need to trade off cognitive foraging costs against navigation foraging costs against information costs -- that are widely relevant to the RTS domain.

These theory-based results reveal opportunities for future explainable AI systems, which in some regards may be based on shoutcasters' explanations, to enable domain experts to find the information they need to understand, assess, and ultimately decide how much to trust their intelligent agents.

The role of explanation systems for intelligent agents in the RTS domain may have been forecast by Shoutcaster Team 3, as they mentioned,

> *"If we took someone who knows literally nothing about StarCraft, just teach them a few phrases and what everything is on the production tab … [then shoutcasters] would be out of a job."*

We hope this work is the beginning of enabling explanation systems to mimic the information provided by shoutcasters and support the information foraging requirements of users.

Bibliography

Bhowmick, S. S., Sun, A., & Truong, B. Q. (2013). Why not, WINE?: towards answering why-not questions in social image search. *Proceedings of the 21st ACM international conference on Multimedia* (pp. 917-926). Barcelona: ACM.

Bostandjiev, S., O'Donovan, J., & Höllerer, T. (2012). TasteWeights: a visual interactive hybrid recommender system. *Proceedings of the sixth ACM conference on Recommender systems* (pp. 35-42). Dublin: ACM.

Castelli, N., Ogonowski, C., Jakobi, T., Stein, M., Stevens, G., & Wulf, V. (2017). What Happened in my Home?: An End-User Development Approach for Smart Home Data Visualization. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 853-866). Denver: ACM.

Cheung, G., & Huang, J. (2011). Starcraft from the stands: understanding the game spectator. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 763-772). Vancouver: ACM.

Chi, E. H., Pirolli, P., Chen, K., & Pitkow, J. (2001). Using information scent to model user information needs and actions and the Web. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 490-497). Seattle: ACM.

Cotter, K., Cho, J., & Rader, E. (2017). Explaining the News Feed Algorithm: An Analysis of the "News Feed FYI" Blog. *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (pp. 1553-1560). Denver: ACM.

Fleming, S. D., Scaffidi, C., Piorkowski, D., Burnett, M., Bellamy, R., Lawrance, J., & Kwan, I. (2013). An Information Foraging Theory Perspective on Tools for Debugging, Refactoring, and Reuse Tasks. *ACM Transactions on Software Engineering and Methodology (TOSEM).* ACM.

Fu, W.-T., & Pirolli, P. (2007). SNIF-ACT: a cognitive model of user navigation on the world wide web. *Human-Computer Interaction, 22*(4), 355-412.

Grigoreanu, V. I., Burnett, M. M., & Robertson, G. G. (2010). A strategy-centric approach to the design of end-user debugging tools. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 713-722). Atlanta: ACM.

Kim, M.-J., Kim, K.-J., Kim, S., & Dey, A. K. (2016). Evaluation of StarCraft Artificial Intelligence Competition Bots by Experienced Human Players. *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (pp. 1915-1921). San Jose: ACM.

Kulesza, T., Burnett, M., Wong, W.-K., & Stumpf, S. (2015). Principles of Explanatory Debugging to Personalize Interactive Machine Learning. *Proceedings of the 20th International Conference on Intelligent User Interfaces* (pp. 126-137). Atlanta: ACM.

Kulesza, T., Stumpf, S., & Burnett, M. (2010). Explanatory Debugging: Supporting End-User Debugging of Machine-Learned Programs. *Visual Languages and Human-Centric Computing (VL/HCC)* (pp. 41-48). Leganes: IEEE.

Kulesza, T., Stumpf, S., Burnett, M., & Kwan, I. (2012). Tell me more?: the effects of mental model soundness on personalizing an intelligent agent. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1-10). Austin: ACM.

Kulesza, T., Stumpf, S., Wong, W.-K., Burnett, M. M., Perona, S., Ko, A., & Oberst, I. (2011). Why-oriented end-user debugging of naive Bayes text classification. *ACM Transactions on Interactive Intelligent Systems (TiiS), 1*(1), 2:1--2:31.

Kuttal, S. K., Sarma, A., & Rothermel, G. (2013). Predator behavior in the wild web world of bugs: An information foraging theory perspective. *Visual Languages and Human-Centric Computing (VL/HCC)* (pp. 59-66). San Jose: IEEE.

Lim, B. Y., & Dey, A. K. (2009a). Assessing demand for intelligibility in context-aware applications. *Proceedings of the 11th international conference on Ubiquitous computing* (pp. 195-204). Orlando: ACM.

Lim, B. Y., Dey, A. K., & Avrahami, D. (2009b). Why and why not explanations improve the intelligibility of context-aware intelligent systems. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2119-2128). Boston: ACM.

Lomas, M., Chevalier, R., Cross, I. E., Garrett, R. C., Hoare, J., & Kopack, M. (2012). Explaining robot actions. *HRI '12 Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction* (pp. 187-188). Boston: ACM.

Metoyer, R., Stumpf, S., Neumann, C., Dodge, J., Cao, J., & Schnabel, A. (2010). Explaining how to play real-time strategy games. *Knowledge-Based Systems, 23*(4), 295-301.

Niu, N., Mahmoud, A., Chen, Z., & Bradshaw, G. (2013). Departures from optimality: understanding human analyst's information foraging in assisted requirements tracing. *Proceedings of the 2013 International Conference on Software Engineering* (pp. 572-581). San Francisco: IEEE.

Norman, D. A. (1983). *Some observations on mental models.* New York: Psychology Press.

Ontanon, S., Synnaeve, G., & Uriarte, A. (2013). A Survey of Real-Time Strategy Game AI Research and Competition in StarCraft. *IEEE Transactions on Computational Intelligence and AI in Games, 5*(4), 293 - 311.

Piorkowski, D., Henley, A. Z., Nabi, T., Fleming, S. D., Scaffidi, C., & Burnett, M. (2016). Foraging and navigations, fundamentally: developers' predictions of value and cost. *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering* (pp. 97-108). Seattle: ACM.

Piorkowski, D., Fleming, S. D., & Scaffidi, C. (2015). To fix or to learn? How production bias affects developers' information foraging during debugging. *Software Maintenance and Evolution (ICSME)* (pp. 11-20). Bremen: IEEE.

Pirolli, P. (2007). *Information foraging theory: Adaptive interaction with information.* New York: Oxford University Press.

Ragavan, S. S., Kuttal, S. K., Hill, C., Sarma, A., Piorkowski, D., & Burnett, M. (2016). Foraging Among an Overabundance of Similar Variants. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (pp. 3509-3521). San Jose: ACM.

Rosenthal, S., Selvaraj, S. P., & Veloso, M. (2016). Verbalization: Narration of Autonomous Robot Experience. *IJCAI.* New York.

Stumpf, S., Rajaram, V., Li, L., Burnett, M., Dietterich, T., Sullivan, E., . . . Herlocker, J. (2007). Toward harnessing user feedback for machine learning. *IUI '07 Proceedings of the 12th international conference on Intelligent user interfaces* (pp. 82-91). Honolulu: ACM.

Vinyals, O., Gaffney, S., & Ewalds, T. (2017, August 9). *DeepMind and Blizzard open StarCraft II as an AI research environment | DeepMind*. (DeepMind) Retrieved 10 12, 2017, from https://deepmind.com/blog/deepmind-and-blizzard-open-starcraft-ii-ai-research-environment/

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering.* Norwell: Springer.

Wong, K. (2014, 10 24). *StarCraft 2 and the quest for the highest APM*. (Engadget) Retrieved 10 12, 2017, from https://www.engadget.com/2014/10/24/starcraft-2-and-the-quest-for-the-highest-apm/

APPENDICES

Appendix 1 - Description of StarCraft II gameplay

StarCraft is real-time strategy (RTS) game that focuses on base building, army building, resource management, and optimization of resource acquisition. The game can be played with basic skills by entry-level players, but also offers virtually unlimited challenges with increasingly complex interrelationships between resources, defensive and offensive play, and overall game strategy limited only by player skill and experience.

The original StarCraft game was released in 1996 and its expansion "Brood War" was released in 1998. Starcraft II "Wings of Liberty" was released in 2010, followed by the "Heart of the Swarm" expansion in 2013 and "Legacy of the Void" in 2015.

StarCraft features three different races, called protoss, terran and zerg. Each race has unique physical attributes, advantages, and limitations, so different game strategies evolve around these unique characteristics. Players are able to choose their representative race at the beginning of the game and players tend to base their race preferences on their game strategies. Players also choose one of many game maps, each with unique geography and resource location.

Protoss are technologically advanced aliens. They rely on an army of robotic ‘probes’ to build their civilization. Strategies employed by the protoss usually rely upon their highly advanced technology and ability to maximize the cost effectiveness of their expensive units. Several key advantages of the protoss include their ability to summon reinforcements to any area they control and personal shields, which must be depleted before an attacker can damage a protoss. Protoss may lose some shield in combat but regenerate quickly. As a result, protoss may also choose to build highly mobile units that can engage in small skirmishes, inflict damage, then retreat before suffering much damage themselves. If protoss get caught out in an extended conflict, though, and unable to retreat, they may be more easily defeated.

Terran are centuries-old descendants of human space colonization. While terran technology cannot match the protoss, they nevertheless have their own special technological advantages. Terran buildings can fly, which allows terran to build infrastructure in a safe area, then rapidly move to another area for resource gathering. This removes much of the risk when expanding into new territory. Terran also command nuclear weapons which makes them a constant threat against any foe. The semi-advanced yet semi-brute-force methods of the terran make them a sort of hybrid option between the protoss and the third race, the zerg.

Zerg are bug-like aliens. They operate in swarms and are ruthless in battle. While zerg lack the technical sophistication of either the protoss or the terran, zerg can overwhelm their enemies with sheer numbers and raw strength. Zerg, however, still have their own nuanced strategies when overwhelming the enemy is not the best option; zerg colonies are built upon 'creep' which is a biological sludge infecting the environment that renders area around the zerg colony unusable to any other race. Spreading creep at resource patches around the map can shut down their opponent's ability to collect resources. Similarly, a zerg may burrow units, some having the ability to explode as living landmines, allowing ambushes from every direction.

A graphic representation of various upgrade sequences is called a technology tree. For StarCraft II, there are a number of different technology trees. Figure A1 shows a simplified technology tree, depicting some possible development paths for the terran race in the StarCraft II "Legacy of the Void." This technology tree was adapted from http://us.battle.net/sc2/en/game/race/terran/techtree/lotv and the full technology tree is available at that website. Technology trees for other races and other versions of StarCraft II are also available.

While gameplay is usually one versus one, there can be up to eight (or even 16) players at a time, limited by map size. There are a hundred or so accepted maps, each with a defined geography and considered "fair" i.e. equally viable for development in all areas. Geography impacts each race differently. For example, protoss, being a high-technology (and high-power, high-cost) race, prefers environments where their

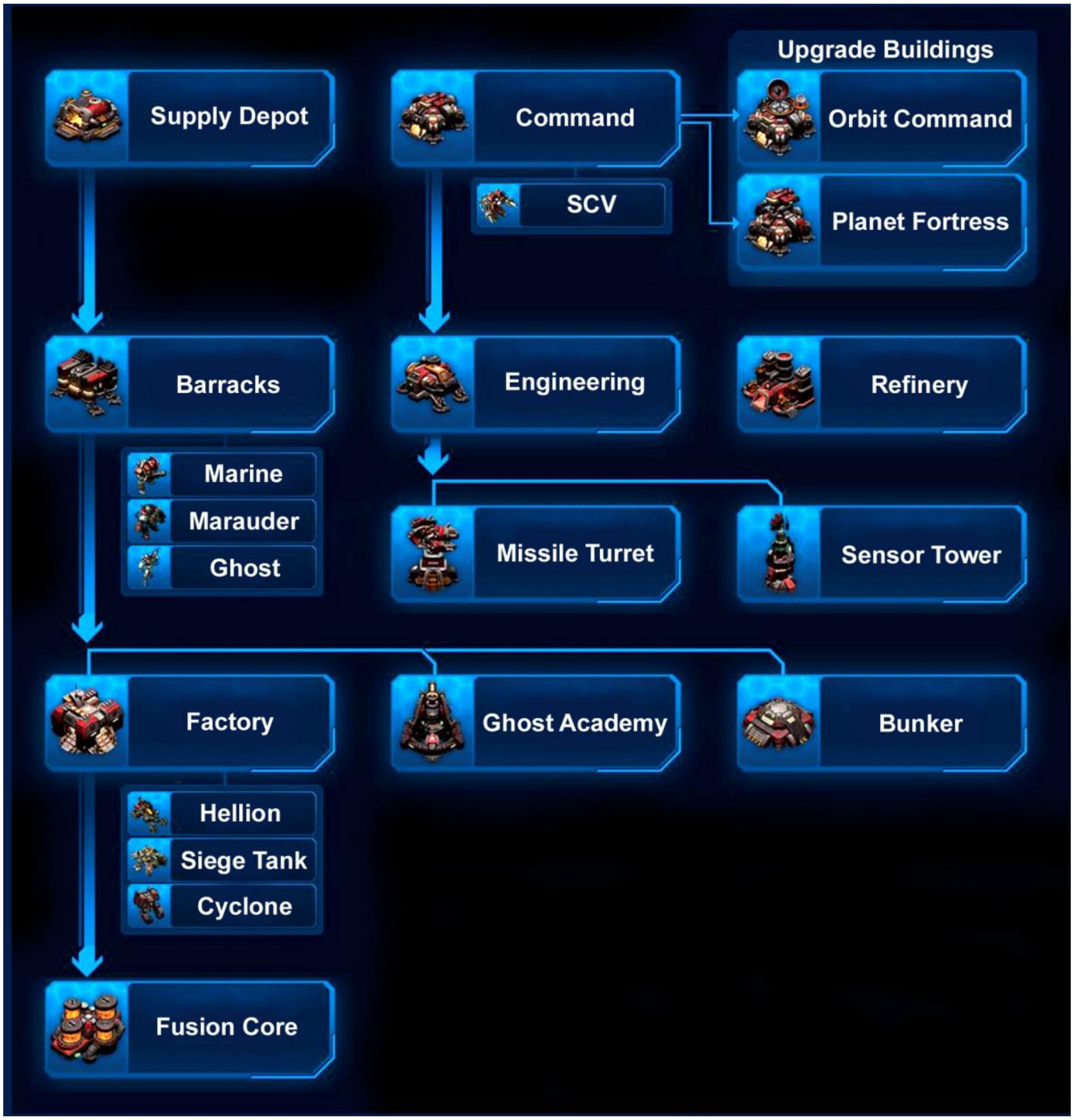

Figure A1. Terran technology tree.

technology is of greatest value. One protoss unit can scale cliffs while another unit can block paths with energy barriers, making more enclosed or mountainous areas a preferable battlefield. Terran units are rather versatile and are not as heavily dependent on cost efficiency as the protoss so may choose a wide variety of geographical locations for their colonies and battlefields. Zerg, as a more swarm-like race, prefers larger open areas where they can surround and overwhelm an opponent. Players can agree on a game map or accept a randomly assigned map. Experienced players take good advantage of prior geographical knowledge.

From the start of the game, players must make crucial decisions regarding their focus, either on increasing their resource gathering capabilities (economy), technology, or military. While all three categories involve distinct allocations of resources, they all crucially interact and determine a player's success. If a player focuses too much on any single aspect, they may be rendered vulnerable if an opponent has intelligence regarding these imbalances. If a player can keep their strategy hidden, they may reap the benefit when they later catch their opponent off-guard.

Unit composition is another key aspect to consider. Players may choose to develop the technology for stealth units instead of offensive units, enabling scouting and sabotage while sacrificing direct engagement capabilities. Likewise, building only land units without investing into air units may leave a player completely defenseless to an aerial bombardment. These tradeoffs dictate a wide degree of opportunity costs for players at every stage of the game.

Players employ reconnaissance units to gather intelligence about an opponent's resources and strategies, but these reconnaissance efforts are limited to opponents' buildings and units that the spies can physically see; spies cannot listen in on opponent's conversations. This reconnaissance limitation leads some players to construct particular buildings to give deceptive impressions of strategies. Further, since reconnaissance units cannot see every building, military buildings in more remote locations might go undetected. Those secret buildings could house construction of an entire secret military force. Gameplay may incorporate a spectacle designed to trick an opponent into believing that a deception is reality.

Taking this counter-intelligence exercise a step deeper in strategy, playing mind games on an opponent might lead them to assume one decision process or strategy, with associated vulnerabilities, while a player actually anticipates and prepares for an expected response to the misinformation. For example, a player might pretend to focus on economic development while actually developing a strong military presence. This could lead an opponent to think it is safe to likewise focus on economic development, only to find that a secretly organized army overwhelms their military weakness. Players might try to keep their strategy hidden while masquerading a plausible facade. This could result in their opponent not being able to react in time, with deception leading to victory. It is therefore crucial to uncover an opponent's deception as quickly as possible. Developing a cohesive strategy thus includes careful consideration of an opponent's likely strategies and their responses to deception.

The multifaceted options for economic, military, and technological development at every moment present a near infinite number of possible scenarios for the player to consider. Small imbalances or weakness in overall strategy may spiral out of control if the opponent has prepared a sufficient counter-strategy. These possibilities thus mirror reality as a constantly developing environment in which every decision has some tradeoff and, inevitably, some result. The ability to balance these necessary decisions and mold those choices into a viable overall strategy represent a solid foundation for any battlefield, virtual or real.