Analyzing Variation of Adaptive Game-Based Training with Event Sequence Alignment and Clustering

Chris Argenta CARGENTA@ARA.COM
Decision Systems Group, Applied Research Associates, Inc., Raleigh, NC USA

Christopher R. Hale CHRIS.HALE@GTRI.GATECH.EDU
Georgia Tech Research Institute, Dayton OH USA

Abstract

Adaptive Game-Based Training (AGBT) systems plan the content and ordering of learning opportunities to customize game-play, thereby addressing individual players' learning needs. Determining the best combination of settings, rules, and algorithms to perform this intelligent tutoring is both complex and expensive, as it requires iteration over multiple experimental cycles so as to measure learning performance and compare it with pre/post testing. In this paper, we propose analyzing event sequence variations produced by the intelligent tutor as an indicator of the effect of changes on adaptive game-play experiences. We describe Event Sequence Alignment and Clustering (ESAC), an analytic method that characterizes variations in the selection and ordering of learning opportunities directly from play-test game logs (without pre/post testing). We present results of a post-hoc analysis of variation, over three experimental cycles, on a large-scale AGBT development effort. We conclude with a discussion of limitations, applications, and future work.

1. Introduction

Adaptive Game-Based Training (AGBT) refers to serious games designed to teach players by evaluating the player's mastery of curriculum concepts and dynamically selecting and adapting sequences of learning opportunities (LO) to meet pedagogical needs. AGBT systems rely on some level of artificial intelligence (AI) to facilitate a player's learning and/or engagement; this is referred to as an intelligent tutor. These tutors can be viewed as cognitive systems that use a model of current player's mastery of curriculum concepts to prioritize training needs, but are constrained by learning theory rules, dependencies between concepts, training time allowed, available game content, and game narrative. These complex interactions make it challenging for development teams to predict how changes to the reasoning process within the tutoring system become manifest in the game-play experience that players receive. Across diverse players, an intelligent AGBT should produce diverse and tailored experiences; however, many serious game studios only evaluate diversity anecdotally, if at all.

The primary measure of an AGBT is learning performance, which is often measured by a pretest-posttest evaluation design. This treats the entire game (including the intelligent tutoring) as a single black-box. Pre/post-testing is effective, but expensive to perform. In the large-scale AGBT project (Heuristica Phase 2) discussed in this paper, only three full experimental cycles were conducted (each guided by 1-2 small-scale pilot tests), however play-testing (subjects playing the game without pre/post-testing) of versions of the game occurred nearly weekly. While play-testing often revealed "bugs" in the tutoring system, analyzing game-logs required manual review and provided little insight into how adaptations changed game-play overall. Because learning performance measures confound changes to the reasoning within the tutor with other improvements in game content, the project could only effectively compare adaptation mechanics on and off: a "static ordering" where everyone received the same (expert specified) sequence of LOs, and the tutor's generated "adaptive ordering" for each experimental cycle. The missing link in this methodology was an ability to directly measure how specific changes to the adaptation mechanics impacted overall game-play experience and the relationship between these interim metrics and expected learning performance. As a result, changes to the tutoring system's algorithm, settings, and input data between experiments often were based on subjective interpretation of learning performance and the team's best guess at what might improve it.

The primary contribution of this research is to describe an analytic for quantifying the range of adaptations exhibited by an intelligent tutoring system within the context of an AGBT, diverse players, and multiple game constraints. We introduce a metric for assessing variation in game-play by identifying similarities within observed sequences of events. Characterizing the amount of variation in game-play sessions allows us to better understand complex relationships between the rules governing the tutor's adaptations and the resulting tailored game-play session. When combined with external performance measures (e.g., pre/post testing) this analysis helps bridge the gap between design and implementation choices, end-user experiences, and learning performance.

In the following sections, we discuss the design and application of an analysis of variation in game-play sessions. First, we introduce our Event Sequence Alignment and Clustering (ESAC) analytic method and describe how it computes the key metrics we propose to support AGBT design. ESAC was developed to analyze observed behaviors of cognitive agents and identify similarities in behavioral sequences. In this work, we applied it to understanding how different versions of an intelligent tutoring performed in the context of diverse players and other constraints. Next, we present a case study of a large-scale AGBT development project, and the post-hoc application of ESAC to existing game logs to explain how the intelligent tutoring system tailored game-play over three experimental cycles. While the original work was conducted with an intuitive sense of how changes might impact game-play, the project suffered from uncertainty in if and how changes would manifest during large scale experimentation. Finally, we conclude with a discussion of limitations of this approach, potential applications, and future work.

2. Analyzing Game Logs with Event Sequence Alignment and Clustering (ESAC)

Event Sequence Alignment and Clustering (ESAC) analyzes observations of activities performed by cognitive agents over time and groups them by similarity for case-based comparisons. When

applied to AGBT systems, we have two cognitive agents being observed: a player and an automated intelligent tutoring system. In previous research (Argenta, Stewart 2014), we introduced ESAC for analyzing and comparing player behaviors. The key questions we wish to answer in this research are: *Given a set of unique and dynamic players and all the fixed game constraints, how much adaptation of game-play experience does the tutor actually produce?* And, *how do changes to the reasoning performed within the intelligent tutoring system (e.g., we change the learning theory approach) translate into changes in the range of resulting experiences?* We answer these questions by retargeting ESAC to analyze sequences of event that are directed by the tutoring system and present in player logs over multiple experiments. This shifts the focus from how players played the game, to how the tutor structured the experience given the player's performance and actions.

Our Log data consist of events observed over time, with each event indicating an event type, a timestamp, and relevant values. Logs are used to provide a persistent record of transient events and relevant state data within a system. Logs are produced during formal experimentation, intermediate pilot testing, and frequent play-testing: making them a valuable source of insight into the evolution of a game.

In this research, we propose the analysis of AGBT logs from experimentation (and play-testing sessions) to evaluate the variation in game-play experiences produced by an intelligent tutoring system, as reflected by the selection and ordering of learning opportunities. Currently, analysis of log data often includes counting events of some type (e.g., how many times something occurred within some period of time) and manual forensic review (e.g., walking through precursor events preceding a problematic event, to understand the situation and behaviors leading up to and perhaps causing the event). Event Sequence Alignment Clustering (ESAC) is unique because it automates the process of extracting key elements of the story being told in the logs as sequences, comparing sequences to each other and grouping them by similarity based on the temporal ordering of their relevant constituent events (Argenta, Stewart 2014). ESAC accomplishes this by performing k-medoid clustering of logs based on the global alignment of the key events.

ESAC is domain agnostic, with all game/log specifics encapsulated in the pattern files, allowing it to be mapped to a wide range of AGBT systems. To expose the tutor's behaviors, we used patterns for each LO scheduled, and for the concept being taught by each LO being played. By performing this analysis on a collection of logs, we can identify a smaller number of "key sequences" (e.g., the "mediod" for each cluster is the sequence with the best average similarity to all other sequences in the cluster) that most generally encompass sequence variations within each cluster. The medoids are the best case sequences to use in classifying a new sequence.

The average of the similarity of all sequences in a cluster to their medoid is called the Intra-Cluster Similarity (ICS). During cluster selection, ESAC attempts to maximize this measure. These "key sequences" also represent the diversity of sequences in the logs. A second important measure affecting variation is the Inter-Medoid Similarity (IMS), which represents the similarity between the "key sequences" of any pair of clusters. A lower IMS indicates two clusters have little similarity. Both IMS and ICS can be inverted to provide a measure of dissimilarity, or variation, between sequences. We propose Total Observed Variation (TOV), the sum of the average variation for each cluster, as a simple measure of diversity of the game-play experiences in the logs analyzed.

2.1 Related Techniques for Sequence Analysis

Many common analytic techniques expressly consider the temporal ordering of events, such as the many variants of parsing/lexical analysis (abstracting meaningful sequences into tokens), finite state machines (classifying acceptance of a sequence), and pattern recognition (including predicting next events in sequence). These, however, focus on matching observed event sequences to known/expected patterns or rules.

Sequence mining techniques are useful for identifying common subsequences and patterns within logs (Marbroukeh Exeife, 2010). ESAC complements these techniques by adding unsupervised learning of groups of similar event sequences. These sequences could be tailored to focus on previously mined common subsequences (not implemented in this research) and/or filtered by regular expressions (implemented in this research). ESAC was created to support plan, activity, and intent recognition (PAIR), where sequences of observable events reflect the behavior of agents (Sukthankar, et. al., 2014). Approaches in this domain include a variety of probabilistic grammar-based algorithms, such as (Pynadath, Wellman, 2000)). ESAC provides a means of comparing observation sequences to a library of sequences in a case-based fashion.

ESAC applies several existing analytic techniques in accomplishing its task. ESAC uses a simple form of parsing to tokenize key events, regular expressions to filter sequences of interest, global sequence alignment (based on Needleman, Wunsch, 1970) to measure similarity, and k-medoid clustering to group sequences (Kaufman, Rousseeuw 1987). ESAC is the product of integrating these techniques for the purpose of comparing and grouping logs by similarity. Similar combinations of sequence alignment and clustering have been developed in the Bioinformatics domain (Corpet, 1988).

What makes ESAC unique is its integrated workflow (Figure 1) and tailoring to log-based event sequences. ESAC includes configurable extraction to generate and validate event sequences from logs databases, automatic pairwise similarity scoring for normalized comparison of diverse sequences, and iterative clustering to converge on the best cases from the log to serve as representatives for each cluster. This combination is well suited for analyzing variation in adaptive systems, such as game-based training, and provides an innovative method for quantifying variation to track the effect of changes to the underlying adaptation mechanics.

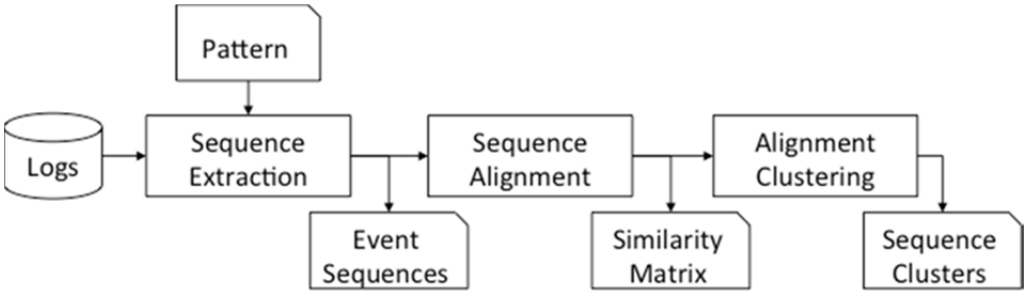


Figure 1. ESAC extracts sequences from a database of logs, computes pair-wise alignment scores, and uses these to cluster similar sequences.

2.2 Methodology for Extracting, Aligning, and Clustering Event Sequences

The ESAC is a three-step process carried out over a repository of logs. The first step, Sequence Extraction, uses a pattern description to produce Event Sequences, which consist of a string-based

token sequence for each unique entity/session in the log. The second step performs pairwise Sequence Alignment on the Event Sequences, and produces a Similarity Matrix indicating how similar each sequence is to every other. The third step, Alignment Clustering, uses the Similarity Matrix to heuristically cluster Event Sequences.

Sequence extraction can be used independently of the other steps in ESAC to characterize log instances against a set of pattern files to extract which players exhibited known behavior patterns. This might be used to produce instance characterization data for machine learning algorithms that then determine what behavior patterns are affiliated with success/failure in the game. However, for ESAC the sequences extracted in this step are fed into the sequence alignment step.

2.2.1 Step 2: Sequence Alignment

Sequence alignment in ESAC consists of pairwise global alignment of all extracted sequences. Higher alignment scores indicate sequences that have more temporally ordered events in common. To compute alignment scores we apply a simplified string alignment implementation using dynamic programming, derived from existing bioinformatics solutions for DNA sequence comparisons. The alignment scoring is based on simple integer credits and penalties that are summed (and later normalized with respect to sequence size).

The algorithm determines the optimal global alignment for each pair of sequences using a method that credits the score +1 for matched tokens and penalizes the score -1 for mismatched tokens and blanks (skipped tokens in either sequence). More complex alignment methods, such as semantically-aware scoring, are possible, but not implemented in the work presented here.

2.2.2 Computing a Similarity Measure from Alignment Scores

Alignment generates a similarity matrix containing normalized alignment scores between each sequence (identical sequences get a score of 1.0). We normalize the alignment scores (in a Similarity Matrix) based on sequence lengths using *Equation 1*, producing a similarity score ranging from -1 to 1 (negative values indicate sequences have more differences than similarities). Normalizing compensates for higher alignment scores from longer sequences.

$$similarity_{AB} = \frac{Score_{AB} * 2}{A_{length} + B_{length}}$$

Equation 1. Formula for similarity between sequences A and B using an alignment score normalized for sequence length.

Similarity provides a standard measure of commonality in the sequential ordering of events between two sequences that can be computed quickly. One important application for similarity measures is for clustering of instances that otherwise have no convenient Euclidian space or dimensions.

2.2.3 Step 3: Alignment Clustering

The third step in ESAC is clustering of sequences based on similarity measures. The k-medoid clustering algorithm is well suited for this, as it does not require a dimensional space for positioning centroids (as k-means does). The k-medoid algorithm is outlined below and creates

clusters around sequence instances using (1.0-similarity measure) as a distance between sequence instances. Alignment clustering performs the following steps:

- **Step 1**: Assign k random medoids (from the set of sequences)
- **Step 2**: For each sequence, assign it to the cluster of the medoid with the highest similarity.
- **Step 3**: Refine within each cluster, by finding the optimal medoid for cluster
- Repeat Steps 2 and 3 until clusters stop improving.

K-medoid clustering (like many clustering algorithms) is sensitive to the initial random selection, but is relatively fast-running. So, we simply sample the initial selection multiple times (more sampling offers more reliable cluster counts) and select the best clustering for each k>1. This is more practical than optimal. Automating the selection of k is performed by incrementally starting with k=1 and increasing until the improvement in the average intra-cluster similarity falls short of a threshold (currently 1%). This balances the trade-off between fewer/larger and more/smaller clusters by finding the point of diminishing returns for adding clusters. The result of the Sequence Clustering step is an assignment of each sequence to a cluster, the medoid for each cluster, the similarity between each sequence and its medoid.

2.3 Intra-Cluster Similarity, Inter-Medoid Similarity, and Observed Variation Metrics

We define are three key metrics to summarize the clustering that results from ESAC:

Mean Intra-Cluster Similarity (μICS) is the average similarity between each sequence and the medoid representing the cluster to which it was assigned. It can be computed per cluster or for all clusters. A low per cluster μICS would indicate sequences in the same cluster were not similar. A per cluster μICS of 1 would indicate sequences that were identical (no variation). ESAC determines the number of clusters by detecting diminishing returns in the μICS across all clusters.

Mean Inter-Medoid Similarity (μIMS) is an indicator of the relative similarity of cluster medoids to one another. The μIMS will always be less than μICS.

Variation is the opposite of similarity, so 1-μICS gives the average variation within the clusters and 1-μIMS is the average variations between clusters. However, there is a trade-off between the number of clusters and these metrics, more clusters results in higher μICS, but with dimishing returns. *Total Observed Variation* (TOV) combines these metrics (*Equation 2*) to estimate an "area" of variation being sampled by the sequences observed.

$$Total\ Observed\ Variation = \sum_{c=each\ cluster} (1 - \mu\mathrm{ICS}_c)$$

Equation 2. Formula for total variation sums the diversity of each cluster.

We propose that TOV represents a simple metric for comparing the diversity of game-play experiences within a collection of game logs. More observed variation indicates that the tutor has leveraged a greater range of adaptations, presumably for tailoring the experience to the player. The tutor's sequence options (potential variation) are still constrained by learning theory, dependencies between concepts, training time allowed, available game content, and game narrative. ESAC also helps the analyst achieve insight into these structural constraints by identifying the key sequences (medoids) and μIMS, directing manual review to a few key game sessions that best represent the variation observed.

3. Application of ESAC to an Adaptive Game-Based Training Case Study

The requirement for metrics to characterize variation in game-play arose from our experience developing and integrating the systems for, and analyzing data from, a large-scale AGBT research project called Heuristica. Heuristica is a serious game designed to teach users to recognize and mitigate cognitive biases (Mullinex, et.al., 2013). In this case study, we focused on post-hoc analysis of logs from Heuristica to characterize variation in game-play experiences over three experimental cycles (referred to as Exp 4, 5, and 6). Heuristica was developed and evaluated prior to our applying ESAC for analysis of variation in game-play, so this research did not contribute to the game, software, or experiment design process. Instead, we started with existing de-identified data, recombined existing experimental conditions and data to focus on the adaptation mechanics, and performed the analysis described here independently of the Heuristica research. For this reason, we include only minimal details of ARA's Heuristica game, GTRI's Student Model (intelligent tutor), and the project's pre/post-test instruments.

3.1 Leveraging Heuristica Phase 2 Log Data

Heuristica is a science fiction game that takes place on a space station, in which the player experiences a series of situations requiring decisions with limited information and under time-pressure. The game is designed to teach participants to recognize and mitigate several categories of cognitive biases. Heuristica was designed to be highly adaptable, and consists of a set of mostly independent learning opportunities (LOs). Each LO consists of an interactive experience within the Heuristica narrative focused on teaching a set of curriculum concepts about each bias. LOs can have ordering dependencies on other LOs to enforce narrative constraints on the game.

Under the IARPA Sirius program a team of researchers led by ARA designed, built, and evaluated Heuristica across a set of experiment cycles. Experimental evaluation included six large-scale multi-site experiments (three in each of two phases) studying the training performance and effects of multiple game-variable conditions with pre/post and longitudinal tests. In this paper, we analyze logs from the second phase of Heuristica (experiments 4, 5, and 6) with the intent of quantifying variations in game-play. We restructured the existing experimental conditions and filtered logs to isolate the method of adaptation, shown in *Table 1*. We eliminated data from several conditions (such as game-play repetition) which significantly affect game-play, and combined other conditions that may have affected learning but not game-play adaptation directly. We also eliminated subjects with incomplete game logs or pre/post-tests.

Table 1. We have distilled the existing experimental data, covering many diverse conditions, to three simple conditions focusing on the manner in which LOs are selected and ordered to create game-play. The number of player logs (n) in each condition is given in parenthesis. We excluded logs for original experiment conditions not mapped to our conditions (e.g., multiple sessions and video training), or had anomalies in game sequences (e.g., did not start or end correctly).

Condition	Description of Adaptation Mechanic	Exp 4	Exp 5	Exp 6
Adaptive Order	Sequence of LOs adapts to player's mastery	✓(91)	✓(109)	✓(83)
Mixed Initiative	Adaptive order with occasional players choice	✓(80)		
Static Order	Fixed ordering of LOs as determined by SME	✓(116)	✓(118)	✓(115)

Under the Sirius program, ARA collected de-identified game logs from experiments conducted at 3 remote University sites through our *SiriusTools-Sync* infrastructure. Pre, Post, and 12-week Longitudinal Tests were administered and automatically scored using a web-based testing tool *SiriusTools-Flow*. In addition to program-specified test scoring, *SiriusTools-Flow* also scored tests addressing each concept in our team's curriculum and used this information to seed the intelligent tutor with initial values of each player's mastery levels. These scoring methods differed primarily in the sub-scales and aggregation of sub-scales. In this post-hoc analysis, we used the scoring from *SiriusTools-Flow* to characterize learning performance because it reflects how the scores were used for adaptation of game-play and was available as part of the existing de-identified log data.

3.1.1 Adaptation Mechanics: Static, Adaptive, and Mixed Initiative

In the Static Order conditions, there was no adaptation: All players in an experiment cycle received the same sequence of LOs. This sequence was designed by a subject matter expert for each experimental cycle, taking into consideration the results and basic summaries of log data from previous cycles. In Exp 4, two static orderings were tested.

The sequence of LOs in the Adaptive Order condition were tailored to the individual player by a GTRI-developed intelligent tutoring control system (Whitaker, et.al., 2013) referred to below as the Student Model (SM). The SM uses a combination of pre-test and in-game scores to estimate a player's mastery of each concept in the curriculum after completion of each LO. After each LO, the SM dynamically suggests the next LOs or (if the player has reached sufficient mastery on all concepts) ends the game. The SM drives adaptation of game-play in Heuristica using a set of simple rules, informed by learning theory, leveraging a database of LOs mapped to curriculum concepts, and re-applied for each LO selection. Improvements were made to the SM rules and supporting data between each experiment cycle.

A Mixed Initiative condition also used the SM, but offered players a choice when multiple, possible next LOs were judged to be of equal effectiveness. The Mixed Initiative condition was used only in Exp 4.

3.2 Extraction Patterns, Clustering Results, and Variation Analysis

We used eight extraction patterns to capture the different aspects of game-play experience across the three experiment cycles. We used two different tokenizing schemes: one that uniquely identified each LO (a repeated token would indicate a replayed LO) and one which identified the concept taught by the LO (a repeated token would indicate the same or different LO teaching the same concept). The n values for each extraction pattern are given in *Table 2*, as are the number of clusters identified when we executed ESAC on each group of extracted sequences.

Table 2. The number of logged game-play sessions for each experimental cycle and the number of resulting clusters for each. Notice the aggregation of experiments results in different clustering.

	All Logs	Exp 4	Exp 5	Exp 6
LO-Based	712 logs	287 logs	227 logs	198 logs
Tokenizing	7 clusters	13 clusters	8 clusters	11 clusters
Concept-Based	712 logs	287 logs	227 logs	198 logs
Tokenizing	11 clusters	7 clusters	10 clusters	10 clusters

The resulting clusters are shown graphically in *Figure 2*. As can be seen, these results demonstrate a reasonable distribution of sequences across clusters. Notice that μICS peaks (at or near 1.0) reflect game-play sessions based on a static orderings (not adaptive) where all sessions follow the same prescribed LO sequences (and therefore concepts).

3.2.1 Anecdotal Observations of Development Changes Reflected in Similarity Clustering

One observation to be drawn from Figure 2 is that, when tokenized by LO, clustering of all the logs indicates that across all experiments there was similarity between the Adaptive Orderings (and potentially Mixed Initiative) and the Static Orderings. This does not appear when isolated by experiment. This suggests that the Static Ordering of later experiments was informed by the Adaptive Orderings of the previous experimental cycle.

In Exp 4, there were many different LO orderings (n=12) compared to a relatively low number of concept orderings (n=6). This conforms to an anecdotal observation that the original mapping of LOs to Concepts in this experiment seemed too liberal. For example, some LOs were mapped to Concepts that were related but not directly part of the instruction provided in the LO. Similarly, the strength of the LO to Concept mapping originally was set to 100% for every mapping. This may have resulted in LOs being scheduled by the SM to teach a concept that might not have been the LO's primary topic. Introduction of this non-primary information might have created interference during learning and increased the cognitive load on learners, thereby decreasing performance (Koedinger and Roll, 2012; Sweller, 1988). The change that this motivated was a significant reduction in the mappings and weights of mappings between LO and Concept

Between Experiment 4 and 5, we observed that the Mixed Initiative SM allowed players to select the next LO from a set of equally prescribed LOs, but that the SM code defaulted to a priority based on the identification of the LO (which was not relevant for this application). Changing the SM to randomly pick from the equally useful LOs, without allowing learners to nominate upcoming LOs, might explain the low similarity scores in non-peak (non-statically assigned) clusters in Experiment 5.

By Experiment 6, the clusters were balanced out in terms of both LO and Concept-based sequences. Experiment 6 with the SM operating was the best performing version of our game for Phase 2 in terms of pre/post-test based learning performance improvement. However, since new and improved content was introduced between every experiment cycle concurrently with changes to the SM, the relative contributions of these two changes been difficult to separate. This post-hoc analysis using ESAC enables us to identify the imbalances of previous experiment cycles and validates that changes leading up to Experiment 6 were positive in that they resolved inconsistencies in selection and ordering of LOs to address Concepts.

3.2.2 Analyzing Variation Metrics

As previously described, Variation is defined as the inverse of Similarity, and Total Observed Variation (TOV) sums the Mean Variation for each cluster. TOV and μIMS are summarized over each experiment in *Figure 3*. Variation in each cluster, by experiment condition: Adaptive, Mixed Initiative and Static Ordering, is shown in *Figure 5*.

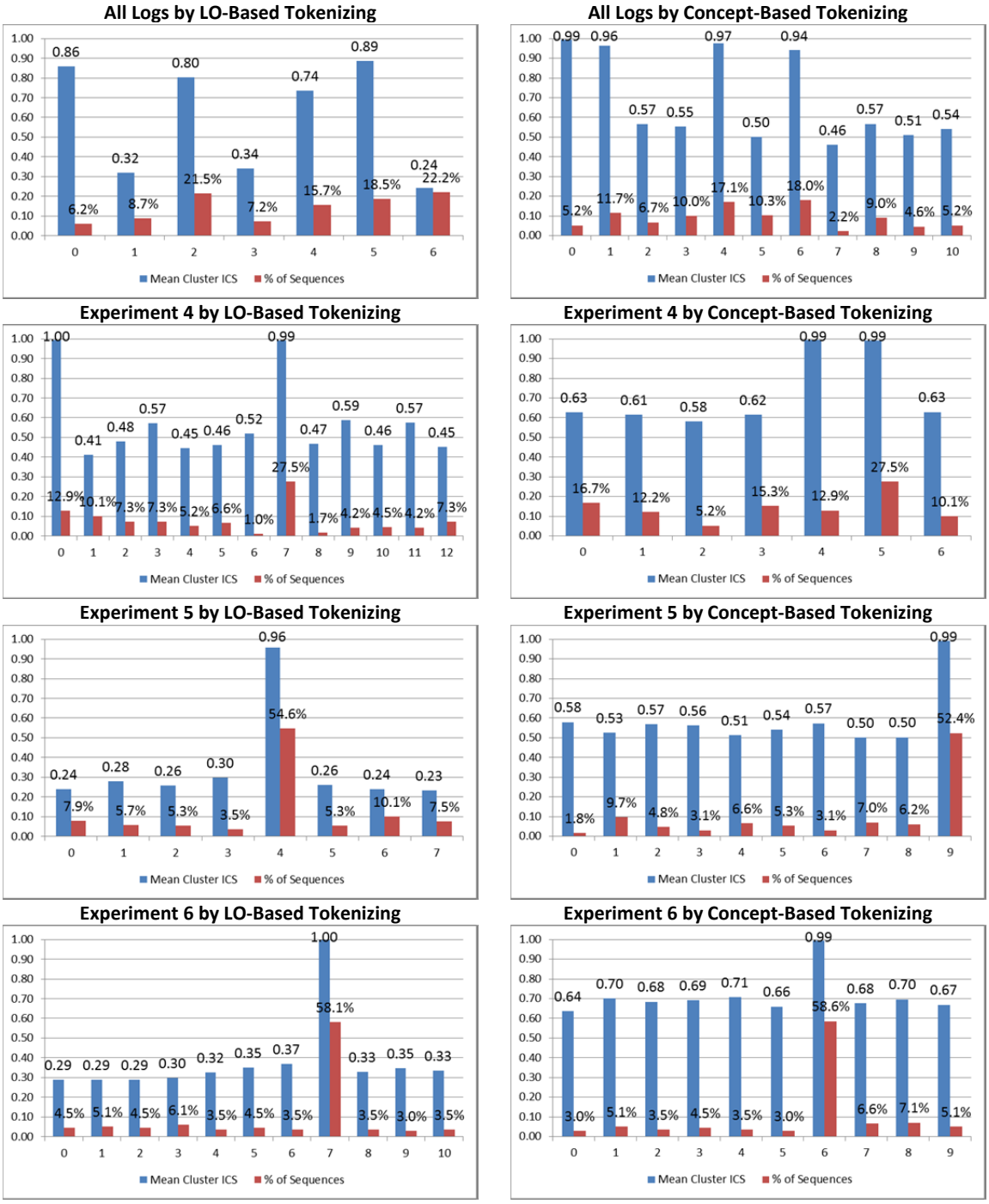


Figure 2. A summary of the clustering for each experiment cycle (and all combined) with different extraction patterns. It shows the cluster μICS and the distribution of sequences (player logs) to each cluster. Notice, clusters with similarity near 1.0 reflect static orderings during that cycle. There is no relationship between the cluster ids (X-axis) between images.

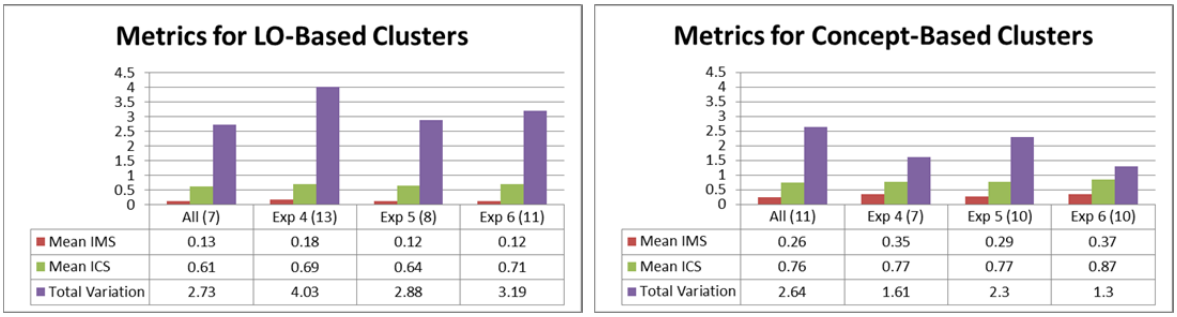
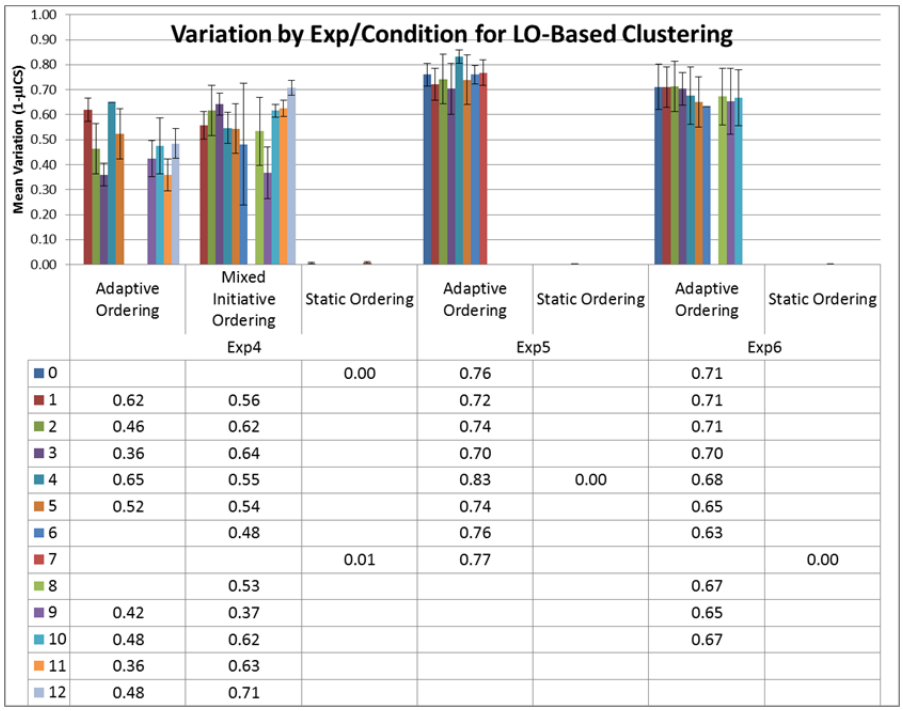


Figure 3. A summary of variation across experiments shows how variation in the game-play sequence has evolved as a result of changes to the SM.

TOV provides one way in which clustering patterns in similarity/variation can be summarized and compared to each other, and is useful in directing attention to differences that may go unnoticed when visually comparing patterns. For example, in the Concept-Based Clustering for Experiment 5 and 6, the patterns in Figure 2 look similar (even have the same number of clusters), but all Experiment 6 clusters have greater similarity than those of Experiment 5. This difference is made clear in Figure 3, as the TOV drops between these cycles. Similarity, comparing the LO-Based Clustering of Experiment 4 with that of Experiment 6 is complicated by a different number of clusters. TOV addresses this and shows that Experiment 6 had less variation (with respect to LO ordering) than Experiment 4.



	Exp4			Exp5		Exp6	
	Adaptive Ordering	Mixed Initiative Ordering	Static Ordering	Adaptive Ordering	Static Ordering	Adaptive Ordering	Static Ordering
0			0.00	0.76		0.71	
1	0.62	0.56		0.72		0.71	
2	0.46	0.62		0.74		0.71	
3	0.36	0.64		0.70		0.70	
4	0.65	0.55		0.83	0.00	0.68	
5	0.52	0.54		0.74		0.65	
6		0.48		0.76		0.63	
7			0.01	0.77			0.00
8		0.53				0.67	
9	0.42	0.37				0.65	
10	0.48	0.62				0.67	
11	0.36	0.63					
12	0.48	0.71					

Figure 4. A summary of variation for each experiment cycle (by lo). similarities. Error bars (stderr of mean) reflect the ranges of variation.

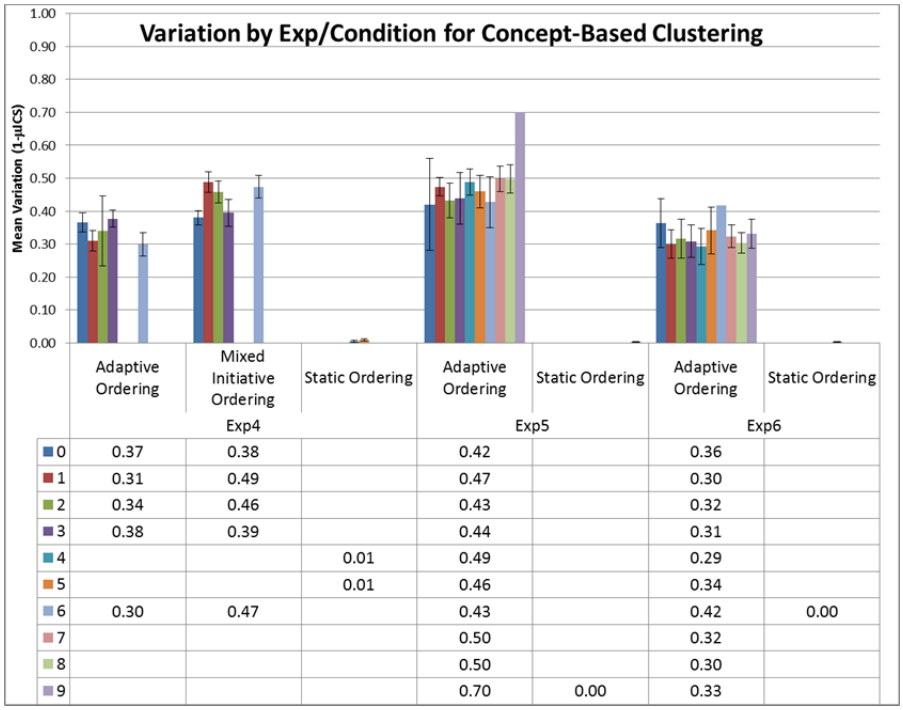


Figure 5. A summary of variation for each experiment cycle (by condition). Different extraction patterns quantifies differences in game-play experience. For example, Exp 4 Adaptive and Mixed Initiative Orderings share considerable similarities. Error bars (stderr of mean) reflect the ranges of variation.

3.2.3 Relationship of Clusters to Pre/Post-Test Learning Performance

The Heuristica project measured learning performance improvement by comparing a pre/post-test (immediate) and pre/follow up-test (at 12 weeks). Figure 4 and 5 summarizes the pretest to posttest improvements for each condition of each experiment, using the concept subscales utilized by the SM. The adaptive ordering led to greater improvements than the static ordering at both immediate and 12-week test intervals.

The most notable finding for learning performance is the reversal that occurs between Experiment 5 and 6. There is a minor improvement in the static ordering condition, which reflects both improvements to the content of the training between these two experiments, and any effects of changes to the static ordering. In the clustering of all logs, all Experiment 5 static ordering sequences fell into cluster 2 (with medoid 50593) and all Experiment 6 static orderings were in cluster 5 (with medoid 60742). These medoids had a similarity of 0.12, suggesting that significant changes were made to the static ordering between these experiment cycles.

There was a major improvement associated with the Adaptive Ordering between Experiment 5 and Experiment 6, consistent with the changes to the SM that improved the balance between LO and Concept-based clustering metrics. The positive effects of achieving this balance are reflected in Fire 5, as well as in the fact that more clusters were associated with improved learning in experiment 6 than in experiment 5 (see Figure 6). In fact, all experiment 6 clusters are above the experiment 5 mean.

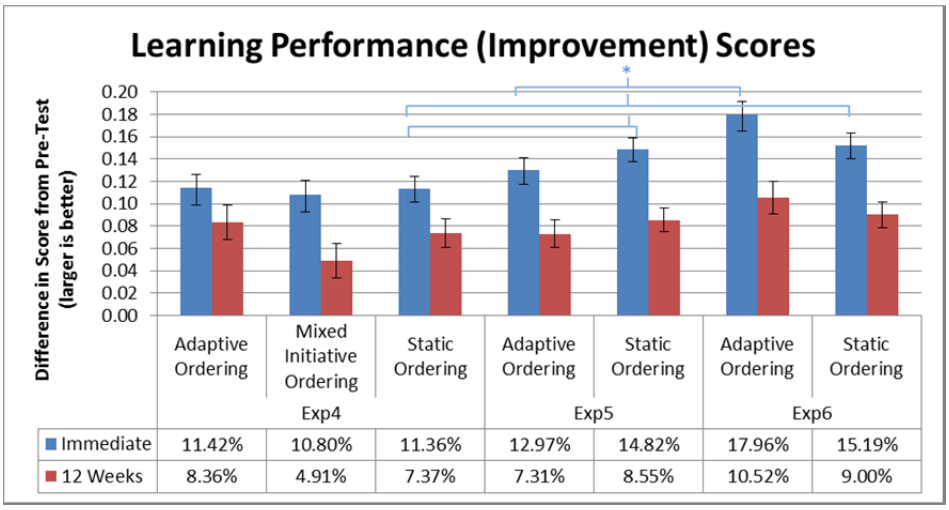


Figure 6. Learning performance (improvement over pre-test scores) increased for each condition over each experiment cycle. (* $p<0.05$)

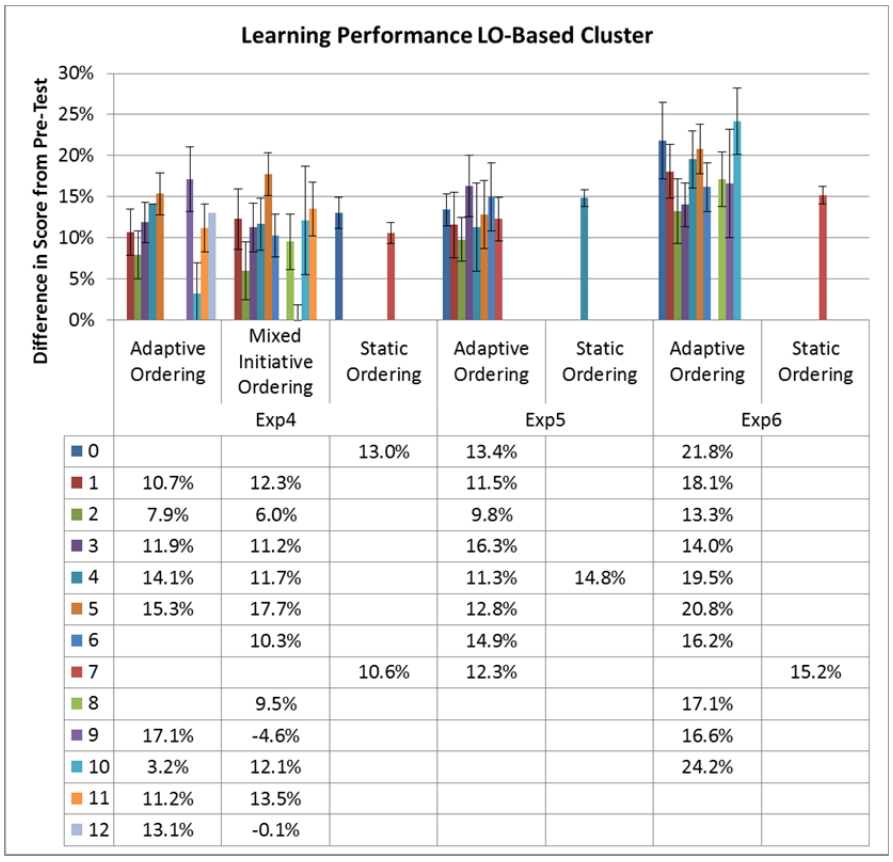


Figure 7. Learning performance by condition and LO-based cluster, shows that the improvement occurred in several clusters.

4. Assumptions, Limitations, and Future Work

Throughout this paper, we have presented variation as a positive attribute of an intelligent tutoring system for AGBT because it allows us to tailor the experience to the individual player. However, simply quantifying variation provides little insight into whether the generated sequences are more or less well suited to each individual. The critical factor seems to be what aspects of the learning environment vary and what aspects are similar. The knowledge components to be learned in any tutoring situation; or in the case of the discussed experiments, concepts to be learned; are most efficiently learned through practice in context. It is well established that much of the process of learning seems to be mediated by focused practice on examples (Matsuda, Cohen, Sewall, Lacerda and Koedinger, 2008; Zhu and Simon, 1987).

A key to such practice is that consistent application of the concepts to be learned be available across a range of examples. In this way, learners are given the practice required to acquire the critical concepts, but do so within a context that facilitates both conceptual stability and transfer of the concepts to new problem solving situations. This suggests, then, that the learning of concepts is best accomplished under conditions of similarity whereas the acquisition of transfer capability is best learned under conditions of example variation. Similarity promotes initial conceptual stability while variation provides opportunities to apply newly learned concepts to a range of problem solving situations, thereby promoting transfer. These conditions were met in the currently reported experiments by disentangling LO presentation from concept learning. The learning improvements across experiments 4, 5 and 6 associated with increasing variation in LOs and similarity in conceptual content supports this prediction.

Thus, more is not necessarily better, as a high degree of variation in the wrong dimension of learning might equate to instability. While variation was shown to be helpful in the structure of the environment within which learning takes place, it was similarity that was important for the learning of concepts.

While TOV is a very simple function, it allows us to compare different clusters in a useful way, and therefore to differentially diagnose the effectiveness of learning situations and to design more optimal learning environments. It is intuitive to interpret TOV as an “exemplar area” covered by learning environments and complements individual cluster μICS measures that focus more on the conceptual content. Combining the concepts of TOV and μICS appears interesting for estimating the space between clusters and may provide a way of relating conceptual similarity and pedagogical diversity in reaching desirable tradeoffs when designing intelligent tutoring systems.

Finally, ESAC was originally designed for use in plan recognition, not log analysis, and is well suited for building hierarchical search trees that limit comparisons to a few medoids at each level. Modifications for log analysis included the concept of pattern-based extraction of sequences. Some of this work may be rolled back into our plan recognition research. For example, we will investigate: applying supervised machine learning to discovering extraction patterns best suited for classifying sequences; using ESAC to cluster synthesized plans to identify plans that are distinctive, and using similar metrics to characterize a planning domain with respect to how easy or difficult it might be to recognize goals.

5. Conclusions

Event Sequence Alignment and Clustering analyzes sequence similarity and variation, producing metrics such as μICS, μIMS, and TOV. A measure of variation helps characterize the diversity of sequences generated by a system, in this case, an intelligent tutoring component within an

adaptive game-based training system. Using these metrics, we performed a post-hoc analysis of three experimental cycles of ARA's Heuristica game-based training for cognitive biases. By configuring extraction patterns, we can focus on different aspects of the sequences, such as as the individual learning opportunities scheduled, or the concepts being taught for each learning opportunity scheduled. We used our analysis of variation to characterize differences in game-play experience that resulted from specific changes to the system over time. We were also able to relate these changes to learning performance improvements.

The primary contribution of this research is the presentation of an analytic method (ESAC) for evaluating sequence variations within game logs and relating those variations to underlying factors important to both learning and the facilitation of transfer. The results presented in three experiments demonstrated that increases in conceptual similarity were associated with improvements in initial learning whereas increases in the variety of learning opportunities led to enhanced transfer. Thus, when designing intelligent tutoring systems, it is important to independently manipulate both the content of the material to be learned and the structure of the learning environment, as represented by learning opportunities (or examples). Manipulating, and measuring, conceptual similarity and learning environment structure independently will allow ITS designers to calibrate and trade these factors so as to achieve the most effective learning and transfer possible.

This method enables researchers to achieve that goal by producing metrics that help characterize the effect of changes to the input data, game content, and intelligent tutoring algorithm on game-play. It can be used with or without pre/post-test, allowing developers to leverage logs from frequent play-testing sessions to help evaluate complex changes that would otherwise require extensive manual review and subjective assessment.

Acknowledgements

The development of Heuristica and the collection of experimental data used in this research were supported by Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory Contract #FA8650-11-C-7177 to ARA, Inc. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, AFRL, or the U.S. Government.

The authors would like to thank Dr. Jon Doyle of the NCSU Computer Science Department who advised the academic work that spawned ESAC and provided feedback to improve this paper.

References

Argenta, Chris, Stewart, Eric (2014), "Extracting Short Stories from Large Data Sets." 1st *Workshop on Human-Centered Big Data Research*, Raleigh, NC,

Corpet, F., (1988) "Multiple Sequence Alignment with Hierarchical Clustering", *Nucleic Acids Research*. Vol. 16 (22).

Kaufman, L. and Rousseeuw, P.J. (1987) "Clustering by means of Medoids", in *Statistical Data Analysis Based on the L_1–Norm and Related Methods*, edited by Y. Dodge, North-Holland, 405–416.

Koedinger, K.R. and Roll, I. (2012). Learning to think: Cognitive mechanisms of knowledge transfer. In K.J. Holyoak and R.G. Morrison (Eds.), The Oxford Handbook of Thinking and Reasoning (pp. 699-716), Oxford: Oxford University Press.

Nizar R. Marbroukeh, C.I. Exeife, (2010) "A Taxonomy of Sequential Pattern Mining Algorithms." *ACM Computing Surveys*, Vol. 43, No. 1, Article 3.

Needleman, Saul B. and Wunsch, Christian D. (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins". *Journal of Molecular Biology* 48 (3): 443–53.

Matsuda, N., Cohen, W., Sewall, J., Lacerda, G. and Koedinger, K.R. (2008). Why tutored problem solving may be better than example study: Theoretical implications from a simulated-student study. In A. Esma and B. Woolf (Eds.), *Proceedings of the Ninth International Conference of Intelligent Tutoring Systems* (pp. 11-121), Berlin: Springer.

Mullinex, Gwen, et.al. (2013) "Heuristica: Designing a Serious Game for Improving Decision Making", *Proceedings of the IEEE International Games Innovations Conference*

Pynadath, DV, Wellman, MP, "Probabilistic state-dependent grammars for plan recognition." *Proceedings of the 16th Conference on Uncertainity in Artificial Intelligence*

Sukthankar, G., Goldman, R., Geib, C., Pynadath, D., Bui, H.H., (2014) "*Plan, Activity, and Intent Recognition Theory and Practice*" Morgan Kaufman, Waltham MA.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. Co*gnitive Science, 12*, 257-285.

Whitaker, E., Trewhitt, E., Holtsinger, M., Hale, C.R., Veinott, E., Argenta, C., Catrambone, R. (2013) "The Effectiveness of Intelligent Tutoring on Training in a Video Game: An Experiment in Student Modeling with Worked-Out Examples for Serious Games", *Proceedings of the IEEE International Games Innovations Conference*

Zhu, X. and Simon, H.A. (1987). Learning mathematics from examples and by doing. *Cognition and Instruction, 4(3),* 137-166.