Preliminary Study of Attention Control Modeling in Complex Skill Training Environments

Heejin Lim[1] and John Yen[2]

[1] Texas A&M University, College Station, TX 77843, USA
hjlim@cs.tamu.edu, jyen@ist.psu.edu
WWW home page: http://people.cs.tamu.edu/hjlim
[2] The Pennsylvania State University, University Park, PA 16802, USA

Abstract. In complex skill-training systems, trainees are required to master multiple skills in a limited time, which may produce a large mental workload. Increased workload often affects performance, and trainees may get distracted or overloaded during training. Attention control is a critical activity in time-sharing environments with automatic tasks, and psychologists found that better attention control strategies can develop through training. Even though attention management is a key skill has to be acquired, it has not been considered to assess as a user model content sufficiently. In this paper, we propose an approach for attention-control modeling by detecting regular behavioral patterns that potentially explain the interdependency between primary and subtask performance. We can detect trainees' attention shift between tasks by interpreting the serial episodes of behaviors that have been uncovered. As a high attention needing training domain, we used Space Fortress game in which continuous input stream of ship maneuvering and intermittent event data are the source of the user model. We found the dependencies between these heterogeneous, multi-time streams and the point of attention shift. Domain experts or training coaches can infer the trainees' attention-control skill based on the detected rules of pattern that help them to instruct desirable strategies to handle multi subtasks.

1 Introduction

A user model consists of all the information and explicit assumptions regarding all aspects of the user. User modeling techniques can effectively support learning on demand by helping users to identify opportunities to learn additional functionality relevant to the task at hand and to prevent people from getting stuck in a sub-optimal plateau. Generally, user models address three issues: (1) inferring the user's knowledge or general abilities; (2) recognizing the user's plans or goals; and (3) predicting the user's inferences and behaviors. The first issue is also called the assessment problem. Assessment has been an important issue because even when it is possible to process numerous observations about a person, with carefully chosen and interpreted references to an extensive empirical

database, the task of dealing with the uncertainty associated with the evidence is challenging [10]. The uncertainty of assessment is unavoidable because of the gap between the observed and the inferred evidence and the conclusions drawn. Recently, user modeling has shifted its focus from dialogue systems to interactive applications. When the user controls interactions with applications such as games or training programs, it is hard to identify relevant information that will become the content of the user model and may later on serve to justify an adaptation step.

Knowledge representation and machine learning technique have been used to build user model. Compared to knowledge representation based techniques, machine learning typically accepts observations of user behaviors rather than assumptions that are statements about the user or already formulated in the internal representation. Machine learning techniques implement some sort of uncertainty by using probability, distance measures, and network weights, etc. From the standpoint of acquisition of user-model content and decision tasks, it has an advantage over observing data and making direct decisions from learning algorithms [15]. The purpose of the user model in training programs is to improve the training performance based on the detection of the trainee's current skills and less-than-optimal strategies. In complex skill training, the substantial numbers of individuals fail to develop proficiency and the performance of an expert is qualitatively different from that of a novice. Trainees are required to master multiple skills in a limited time, which may produce a large mental workload. The difficulty is that some trainees can perform an individual component skill well, but cannot operate well in high-workload situations [17]. The attention-control skill is defined as the ability to allocate attention away from an automatic task to concentrate more resources on non-automatic tasks. According to cognitive scientists [22, 6], we can perform multiple tasks at one time, as long as we have enough attentional resources and the tasks do not interfere with each other.

In psychological perspective, trainees usually move from explicitly controlled processes into implicit, automatic processes [18, 16]. Therefore, the author hypothesizes that trainees' attention control skill can be assessed by modeling automaticity of the primary task, while concurrently manipulating the secondary tasks. In other words, as the skill in the primary task increases, available attention to handle secondary tasks will increase, which enlarges reserve capacity, so that trainees can positively respond to secondary tasks. In this research, we propose an approach for attention-management modeling using Data Mining techniques. We apply rule-discovery methods aimed at finding relationships from the multiple time series with modified episode rules [13]. By detecting regular behavioral patterns that potentially explain the interdependency between primary and subtask performance, we can assess trainees' expertise in attention control and map out the attention shift between tasks.

2 Related Work

Finding out hidden rules that can represent dependency between one time series data and the other one is interesting problem. The association discovery rule

[2] has been developed for finding coexistence of certain values together from a set of transactions. The AprioriAll algorithm [1] can operate on categorical data and considers the coexistence of items within a range. Many applications have used this algorithm to detect association or sequential patterns in single time series data [7, 21]. However, the effort to handle multiple, heterogonous time sequences with concurrent events has been tried only rarely. Researchers have tried to discover temporal patterns from multiple streams of homogeneous time series. MSDD (multi-stream dependency detection)[14] method uses a general-to-specific algorithm to detect significant dependencies between multiple streams. It treats dependency detection as an efficient search for the k most predictive rules in search space but it has some limitation to a small number (< 6) of categories in time-series data. Because it is sensitive to time interval it is inappropriate for detecting frequent serial events regardless of intervening events. Höppner's method [9] considers the time interval and the order of events. He tried to discover temporal patterns in a single series of labeled intervals. He mapped the continuous time series into attributed intervals to apply temporal logic to the single time state sequence. In fine-grained domains such as Space Fortress game, most events have relatively short durations that have no significant meaning for intervals, it gets complicated when try to map into time intervals. Das et al. have transformed continuous time series into tractable, discrete sequences by using clustered windows [3]. They extended their framework for two series by merging them into a single, discrete sequence. In their work, they did not sufficiently address the problem of how to order clustered data from different time streams into one sequence when the temporal order should be considered.

As another approach to find association of behavioral patterns, Subramanian et al. studied in tracking the evolution of human subjects' control policies from a large, sequential corpus of low-level data [20]. They partitioned episodes into nearly stationary segments using Kullback-Leibler (KL) divergences to track when the change in policy between adjacent segments is significant. The learning model of control policies is a lookup table that contains a distribution of the actions that were taken in response to each observed perceptual input vector in the episodes. However, the lookup table model needs a more compact representation of the detected model that can be conveyed to human trainees.

For multimodal human-computer interaction, inferring the focus of human attention has been recognized recently as an important factor for the effective communication and collaboration between human and computer. Horvitz et al. tried to model human attention such as sensing and reason [8]. They employed a Bayesian network (BN) for inferring attention from multiple streams of information, and for leveraging this information in decision-making under uncertainty, such as an attention-sensitive alerting system. They modeled attentional focus in a single period with many nodes that link to a focus variable with probabilistic value. One limitation of BN is that the system designer should define variables that possibly affect the attentional focus. However, if the network is densely connected, then inference in the network is intractable. In [19] an approach for modeling attention focus of participants in a meeting via an hidden Markov

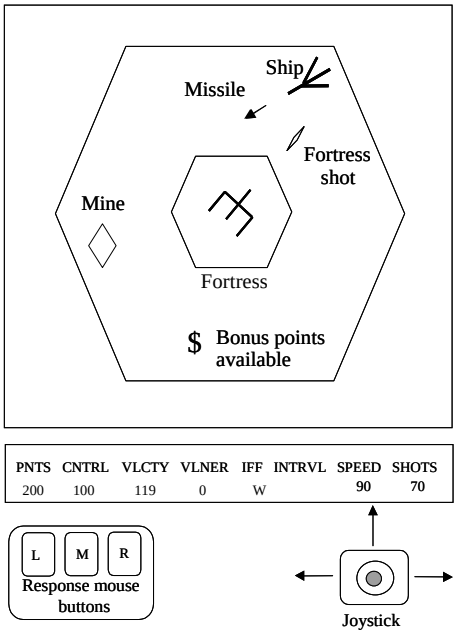


Fig. 1. A Schematic diagram of the Space Fortress game

model was presented. They detected and tracked all participants around the table, estimated their gaze direction by neural network, and mapped the observed gaze to the likely target using a probabilistic framework.

3 Problem Domain: Space Fortress

The Space Fortress (SF) game was developed at the Cognitive Psychophysiology Laboratory of the University of Illinois to simulate a complex and dynamic operational environment [12]. This paper used the ongoing version for distributed team-training SF game as shown in Fig.1. SF game requires a highly concurrent and coordinated use of perceptual and motor skills. Positive transfer from SF training to flight training for U.S. Army helicopter pilots and Israeli Air Force pilots supports the representative nature of the version developed by Gopher et al. [5]. Based on Frederikson & Whites' analysis [4], the SF task can be divided into three goals: (1) circumnavigating the fortress while destroying it as often as possible, (2) handling mines properly, and (3) gaining bonuses. These three goals must be achieved for a high total score. The total score is based on the summation of four sub-scores: Speed (how fast the ship response to friend and foe mine respectively) , Velocity (how low can the ship speed maintained), Points (how many times the ship destroyed fortress and mine or damaged by fortress and mines), and Control (how well the ship maneuvered between the two hexagons) . We can consider that the primary task is to navigate between two hexagons at

low speeds and the secondary tasks are to handle mines, to earn bonuses, and to destroy the fortress. Our approach separates destroying the fortress from the primary goal because we want to classify subtasks according to their relative attentional component.

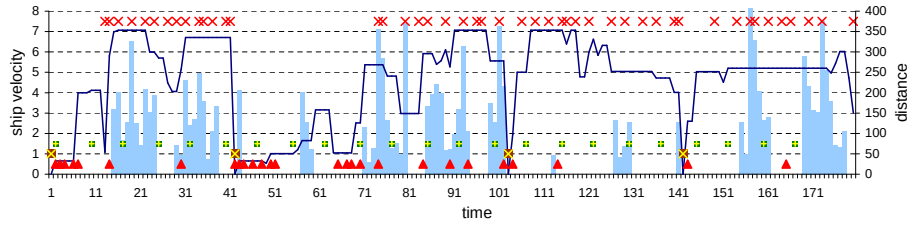
Because the SF system accepts concurrent motor actions from various input devices, it is hard to obtain each skill assessment from multiple continuous stream of performance that shows significant fluctuations over time. We traced the changing of domain states and occurrence of events by generating a history file that contains a long sequence of time-indexed data sets of mixed continuous time series and various other types of events. And then extracted the sequence of skill performance data to find out whether the alteration of primary task performance is related to other subtasks because the performance changing patterns can provide important information about the trainees' attention shift. To see the changing of primary skill performance, i.e., ship velocity, corresponding to secondary skills, let us examine examples of a contingency analysis of SF. Fig.2 illustrates ship velocity change, distance between ship and mine when mine appears, appearance of bonus related characters, ship re-positioning, fortress shell creation, and ship's wrapping events. In this paper, we only consider interdependency between ship velocity skill and mine existence. Fig.2(a) shows that subjects with low level skill showed high ship speed during trials. We can observe that, even when there is no mine, ship speed was still very high because the subject has such a low skill of ship control that the existence of mine did not affect the change of speed significantly. Fig.2(b) shows an example of moderately skilled subjects. Notice that this contingency highly seems that the speed rises rapidly when mines appear and significantly decreases after mine. We find that speed fluctuation is more remarkable than low skilled subjects. Fig.2(c) shows the result from highly skilled subjects. In general, they display low ship speed during trials even when mines appear. Moreover, we can find some regularity that explains dependencies between primary and secondary tasks, as well as individual differences. We can glean these valuable hidden rules from multiple time streams by data mining. For example, a discovered dependency rule is maybe "appearance of mine is followed by increasing of ship velocity with confidence 0.5".

4 Attention Control Model

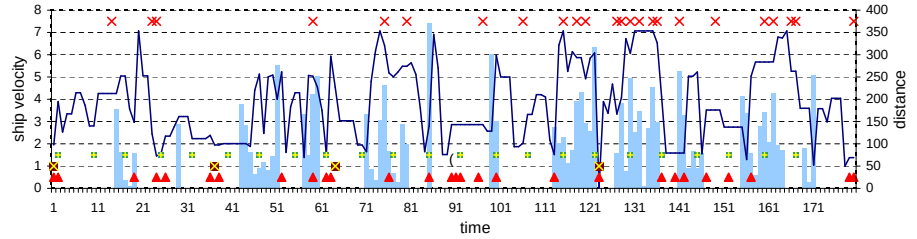
4.1 General Framework

Given multiple time series of heterogeneous types, an attention control model can be constructed. Fig.3 shows an example framework. The following procedure provides the overall sequence, and each process is described in detail below:

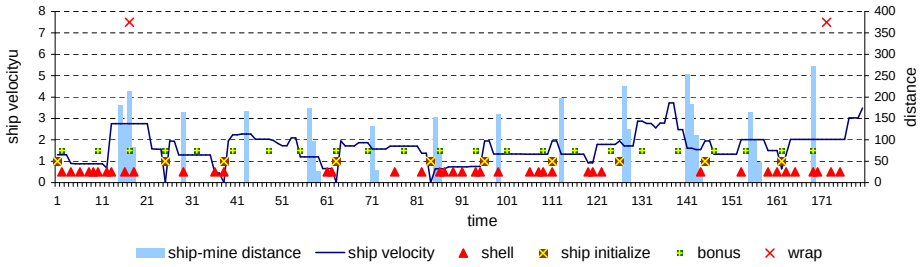
1. Transform the continuous time series into a sequence of categorical data.
2. Transform occasional event tuples into an event sequence.
3. Transform multiple-event sequences into single-event sequences with concurrent event mapping.



(a) Velocity score: -658



(b) Velocity score: -294



(c) Velocity score: 1272 (time span = 1.0 sec., # of time series = 180)

Fig. 2. Contingency examples of Space Fortress game

4. Apply a serial-episode detection algorithm.
5. Generate rules from detected episodes.
6. Filter the generated rules with informativeness measurement.

Transforming continuous time series into a sequence of categorical data

Through transformation of continuous time series into a sequence of qualitative data, we can extract knowledge from a time series that has fluctuations. We can reduce the continuous time-series data into a compact sequence of knowledge including a primitive shape of the pattern. We segmented the time series into subsequences using a *sliding time window* then clustered these subsequences using a suitable measure of pattern similarity method, e.g., K-means clustering,

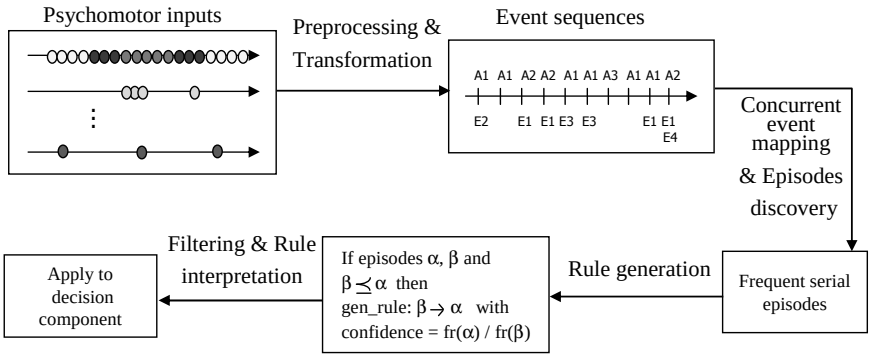
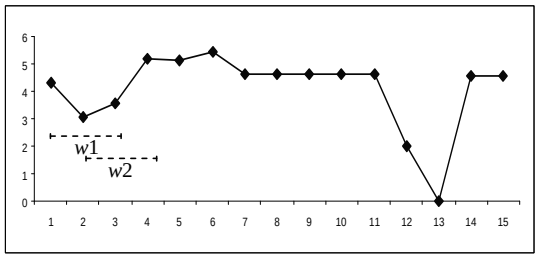


Fig. 3. General framework for modeling attention control

and then assigned categorical symbols to the fragmented subsequences that correspond to the clustered value.

These steps provide a useful method for discretizing time series [3] and bring up several related issues. First, how should the width of the sliding time window be decided? The window size depends on the time scale the user is interested in. A simple strategy for determinizing width w is that we do not consider the size at first, but after detecting and interpreting rules, adjust the size according to the informativeness of extracted rules decided by domain experts. Here, we set $w = 3$ after comparing results of various window size; it reflects well any change of velocity with a fine-grained time interval. Second, how should the time series data in a window be represented for further analysis? In this study, the change of speed is the main knowledge to trace so that we define the distance function as the slope of speed change between the start point and the end point of the window. For example, a subsequence $< x_i, x_{i+1}, \cdots, x_{i+w-1} >$ has slope $s_i = \frac{x_{i+w-1} - x_i}{w-1}$ and the distance function between subsequence x and y is $d(x, y) = \sqrt{(s_x - s_y)^2}$. By using the degree of change of speed as distance functions, even if average velocity values differ from subject to subject, we still cluster the change of speed in a user-adaptive way. The third issue is, if we use a clustering method to process the data, how many clusters would be reasonable, e.g., choose the value of k for K-means clustering algorithm. We chose five with expectation to group into rapid increasing, slow increasing, no change, slow decreasing and rapid decreasing of velocity.

We have time series data X collected at every time step t with time span ts, such as $X(t) = \{t_i | i = 1, 2, \cdots, n\}$ where n is a time index and $t_{i+1} - t_i = ts$. The value of x collected at t_i varies along the time line. A sliding time window of width $|w|$ on time series X is a contiguous subsequence $s_i = \{x(t_i), x(t_{i+1}), \cdots, x(t_{i+w-1})\}$. From a sequence of X, we get the set of subsequences, s_1, s_{n-w+1} of width w, when window slides left to right by one unit (i.e., window movement $v = 1$). The subsequences of continuous time series are grouped by five centroids as shown in Fig.4 when we set $k = 5$.



Clusters	Centroids
A	1.8833
B	1.1059
C	0.6082
D	-1.5818
E	-0.0941

sliding time window (w=3)
#windows = n – w +1 (15-3+1=13)

Time index	< t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15} >
S	< 4.30	3.07	3.54	5.18	5.11	5.44	4.60	4.60	4.60	4.60	4.60	1.99	0	4.54	4.54 >
TS	< e	b	c	e	e	e	e	e	e	d	d	b	a >		

Fig. 4. Transformation of continuous time series by clustering

Transforming occasional events into an event sequence

We have several event types, which occur intermittently and concurrently, such as mouse inputs to take bonus and joystick inputs to shoot mine while continuously manipulating joystick to fly ship. Consider k event types $E = \{e_i | i = 1, \cdots, k\}$. For a given event type $e_i \in E, e_i$ has a time step t_i, the time when the event occurred. An event of type e_i can be represented as a pair $(e_i \in E, t_i)$. For example, an event e_1 is input from mouse device by clicking the left or middle button. Let ES_1 is a discrete event sequence for e_1, which represent e_1, which occurred at time stamp t_i. By using the same sliding time window as in the previous continuous time series, the event sequence ES_1 can be transformed to the modified event sequence MSE_1, where $|MES_1| = n - w + 1$. Being different from transformation of contiguous time series, we do not need a clustering procedure. That is, if event e_i occurred in the middle of subsequence (i.e., $x(t_{\frac{i+w}{2}}) = 1$, when $s_i = \{x(t_i), \cdots, x(t_{i+w-1})\}$) then assign the symbol e_i to the subsequence; else assign 0. This process transform the original event sequence into an equal number of data sequences as that of the transformed categorical sequence.

Discovery of interdependency episodes

We can detect the frequent combination of events by modifying episode rule [13]. First, in the interest of having information about concurrent events, we convert a set of events on the same time stamp into a new event type as shown in Fig.5. Thus, given n types of events and k clusters of continuous data we may attain at most $k \times \sum_i {}_n C_i = k \times (2^n - 1)$ concurrent events type. An episode is a collection of events in a particular order occurring within a given time window. There is a combinatorial explosion problem when n gets larger; however in training domains, we usually have few subtasks to handle. Thus, we do not investigate the combinatorial issue here. Because events from different sequences contain performance information for the corresponding task, the event set that occurs at the same point in time should be considered as a specific event that requires

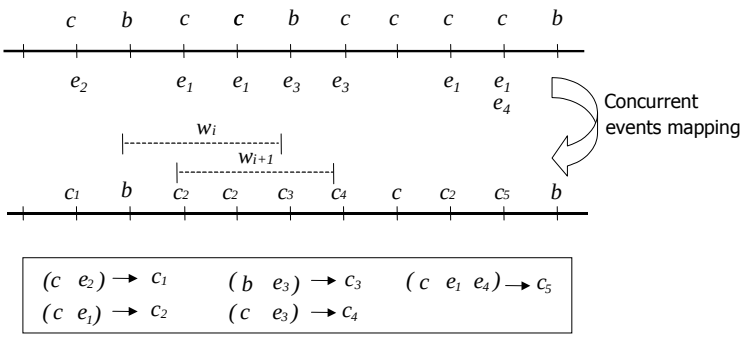


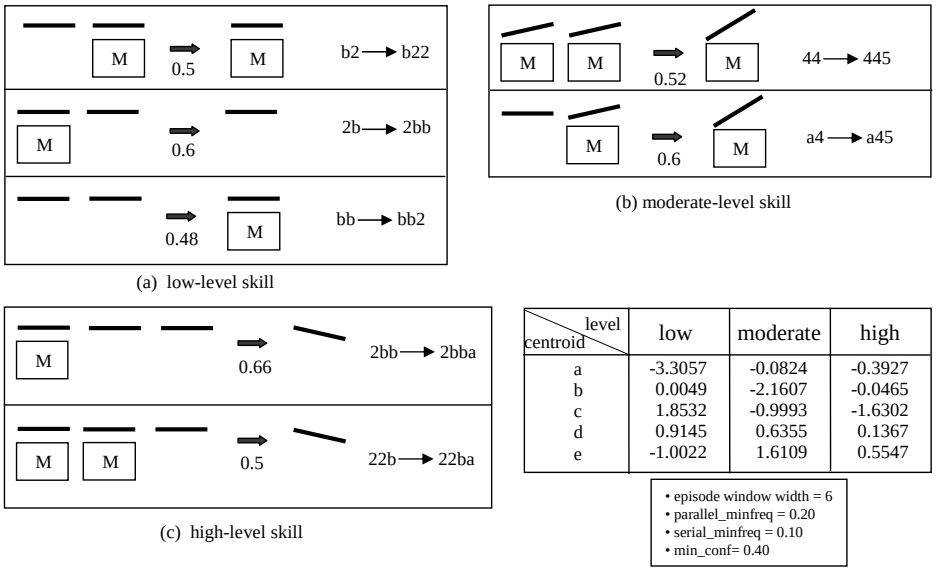
Fig. 5. The example event sequence and two windows

the trainees attention control. A modified serial-episode detection algorithm is a promising method when the order of events in a fine-grained time period has significant meaning. From parallel episodes, serial episodes are detected by recognizing the order of events in windows by screening with a minimum frequency (i.e., min_frequency). This is useful when the size of the sequence is small so that the scanning time is trivial for recognition.

//Serial episode detection algorithm
Compute $C_1 = \{\alpha \in \epsilon \mid |\alpha| = 1\}$
$l = 1$
While $C_l \neq \varnothing$
//compute frequent episodes
$F_l = \{\alpha \in C_l \mid freq(\alpha, s, win) \geq min\_freq\}$
$l = l + 1$
$C_l = \left\{\alpha \in \epsilon \mid |\alpha| = 1, \forall \beta\beta \prec \alpha, |\beta| < l, \beta \in F_{|\beta|}\right\}$
For all l do
// find serial episode from F_l
$FS_l = \{\alpha \in F_l \mid freq(\alpha_i, s, win) \geq min\_freq, i \leq l!\}$
For all $l, l \geq 2$ do
For all $j, j \geq l + 1$ do
If $\beta \prec \alpha$ where $\beta \in FS_l, \alpha \in FS_j, \frac{freq(\alpha)}{freq(\beta)} \geq min\_conf$
// generate serial rules
$\alpha \rightarrow \beta$

Generating rules from episodes

Rule-generating algorithm filters uncovered sequential episodes into serial rules. An objective informative measure, confidence, is used to filter out uninformative rules from the generated rules, providing uncertainty of assessment with statistical methods. This basic approach does not consider any domain knowledge, and as a result, it can generate many irrelevant rules or miss important rules. In Klemettinen et al. [11], rule templates specifying the allowable attribute values



centroid \ level	low	moderate	high
a	-3.3057	-0.0824	-0.3927
b	0.0049	-2.1607	-0.0465
c	1.8532	-0.9993	-1.6302
d	0.9145	0.6355	0.1367
e	-1.0022	1.6109	0.5547

- episode window width = 6
- parallel_minfreq = 0.20
- serial_minfreq = 0.10
- min_conf= 0.40

For low skilled subject's case (a), three serial episodes were detected. As shown in centroid-level table (right below), 'b' represents almost no changing of ship velocity. '2' represents a concurrent event in which behavior 'b' and appearance of mine occurred together. The rule $b2 \rightarrow b22$ explains that in a given time window (6 second in here), if the ship velocity does not change in a while and then mine appears, we can expect that the ship will be in the same velocity with mine appearance in the next step with 50% confidence. (b) and (c) shows detected rules for moderate and high skilled subjects with the same manner as (a).

Fig. 6. Serial episode rule examples

are used to post-process the discovered rules. Srikant, Vu, and Agrawal [19] used boolean expressions over the attribute values as item constraints during rule discovery. As a second filter, training coaches or experts can detect some useful rules from previously filtered rules.

Fig.6 illustrates the largest serial rules extracted from the data of the three trainees shown in Fig.2. For instance, the first discovered association rule in Fig.6(a) shows that when we observe event b and 2 sequentially for the low skilled subject then we expect event 2 to occur, etc. Notice that other events can intervene within these frequent events in the given window. Rules in Fig.6(a) tell us that before the appearance of mine and after disappearance of mine there are rarely change of speed. We can hypothesize that the subject did not shift attention to mine because he/she cannot allocate attention away from primary component that is not yet automated. In Fig.6(b), the rules extracted from the data of moderate-skilled subject show that speed change increases after mine occurs. This implies the mine event distracted the subject because the subject paid attention to mine handling. However, due to limited automation of the primary component (i.e., navigation), the subject's performance for the

component is degraded. Fig.6 (c) shows the rules extracted from the data of high skilled subjects. The rules indicate that the speed of ship tends to decrease a little after mine appearance. This may suggest that the subject's speed control skill is well automated, and hence is not disturbed by the appearance of mine.

4.2 Extensions

The ship-mine distance plot in Fig.2 can be used to extract some high-level behaviors in the context of the game such as let the mine approaches the ship closer and then destroy the mine when it is close enough without changing the speed of the ship, or navigate the ship away from the mine to avoid it. The proposed approach can model these behavior patterns by adding a preprocessing task such as calculating the distance between ship and mine for every time step to get a continuous time series. Through transformation, we can get an event sequence that includes distance-changing information and then apply a dependency-detection algorithm to uncover the dependency rules between ship velocity and the ship-mine distance sequence. One challenging issue is applying these rules to the other user-modeling components for coaching purpose. A promising approach is to feed the rules to the systems decision component to generate feedback. By formalizing part of the rules, a KR-based decision component could generate user-adaptive feedback such as directing the user's attention toward certain events, if the events seem significantly linked to performance, and the user does not respond in the right way. However, this should be considered on the premise that training experts or cognitive psychologists analyze the discovered patterns and rules.

5 Conclusion

In complex skill training environment, the degree of users' attention control skill is one of the most important assessments that should be modeled in an appropriate manner. We developed a preliminary computational framework for modeling attention-management as new user-model content by presenting a dependency model capturing the dependency relationship between the trainee's performances in handling multiple skill components. Inspired by the successes of association detection technique of data mining area, we used a modified algorithm to process raw time-series data and discover hidden rules from multiple streams. The traced data are time-indexed streams of psychomotor inputs received from multiple devices and state information resulting from trainees' behaviors. Provided that continuous, multiple-time series exists and can be correctly uncovered, the dependency rules will trace attention shift between skill components. Furthermore, concerning performance fluctuations, serial episode rules will explain to some extent which events cause or follow performance change. For the further study, not only for an individual trial, we need to model the changing of attention control capability as trials go on. The increasing rate of attention control skill would be a useful user model content.

Acknowledgements

The authors would like to thank Richard A. Volz and Wayne Shebilske for helpful discussion. We would also like to thank Yoonsuck Choe for his helpful comments. This research was supported by AFOSR MURI.

References

[1] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A. I.: Fast Discovery of Association Rules. Advances in Knowledge Discovery and Data Mining, MIT Press Chap. 12 (1996) 307–328

[2] Agrawal, R., Imielinski, T., and Swami, A.: Mining association rules between sets if items in large databases. In: Buneman P., and Jajodia, S.(eds.): Proceedings of ACM SIGMOD Conference on Management of Data Washington D.C., USA:ACM (1993) 207–216

[3] Das, G., Lin, K., Mannila, H., Renganathan, G., and Smyth, P.: Rule discovery from time series. Int. Conf. KDDM (1998)

[4] Frederiksen, J.R., White, B.Y.: Principled task decomposition. Acta Psychologica 71 (1989) 89–146

[5] Gopher, D., Weil, M., and Bareket, T.: Transfer of Skill from a Computer Game Trainer to Flight. Human Factors 36(3) (1994) 387–405

[6] Gopher, D, Weil, M., and Siegel, D: Practice under changing priorities: An interactionist perspective. Acta Psychologica 71 (1989) 147–178

[7] Harms, S., Li, D., Deogun, J., and Tadesse, T.: Efficient Rule Discovery in a Geo-Spatial Desicion Support System. Proceedings of the Second National Conference on Digital Government (2002) 235–241

[8] Horvitz, E., Kadie,C.M., Paek, T., and Hovel, D.: Models of Attention in Computing and Communications: From Principles to Applications. Communications of the ACM 46(3) (2003) 52–59

[9] Höppner, F.: Discovery of Temporal Patterns Learning Rules about the Qualitative Behaviour of Time Series. Lecture Notes in Artificial Intelligence 2168. Springer Freiburg, Germany (2001) 192–203

[10] Jameson, A.: Numerical Uncertainty Management in User and Student Modeling: An Overview of Systems and Issues. User Modeling and User-Adapted Interaction 5 (1996) 193–251

[11] Klemettinen, M., Mannila, M., Ronkainen, P., Toivohen, N., and Verkamo, A. I.: Finding interesting rules from large sets of discovered association rules. In Proc. 3rd Int'l Conf. on Information and Knowledge Management, Gaithersburg, Maryland (1994) 401–408

[12] Mané, A., Coles, G. H., Wickens, C. D., and Donchin, E.: The use of additive factors methodology in the analysis of skill. Proceedings of the Human Factors Society-27th Annual Meeting (1983)

[13] Mannila, H., Toivinen, H., and Verkami, A.I.: Discovery of frequent episodes in event sequence. Data Mining and Knowledge Discovery 1(3) (1997) 259 – 289

[14] Oates, T., Firoiu, L., and Cohen, P.R.: Clustering time series with hidden Markov models and dynamic time warping. IJCAI-99 Workshop on Neural, Symbolic and Reinforcement Learning Methods for Sequence Learning (1999)

[15] Pohl, W. and Nick, A.: Machine Learning and Knowledge Representation in the LaboUr Approach to User Modeling. Proceedings of the 7th International Conference on User Modeling. Banff, Canada (1999) 188–197

[16] Rasmussen, J.: Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering. New York : North-Holland (1983) 136–140
[17] Schneider, w.: Training high performance skills: Fallacies and guidelines. Human Factors 27 (1985) 285–300
[18] Shebilske, W. L., Goettl, B. P., and Regian, J. W.: Individual and group protocols for training complex skills in laboratory and applied settings. In: D. Gopher & A. Koriat (eds.): Attention and Performance XVII: Cognitive regulation of performance: Interaction of theory and application. Cambridge, MA: MIT Press (1999) 401–426
[19] Srikant, R.,Vu, Q., and Agrawal, R.: Mining association rules with item constraints. In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining. Newport Beach, California, AAAI Press (1997) 67–73
[20] Subramanian, D., and Siruguri, S.: Tracking the evolution of learning on a visualmotor task. Technical Report TR02-401, Department of Computer Science Rice University (2002)
[21] Sun, R. and Peterson, T.: Autonomous learning of sequential tasks: Experiments and analysis. IEEE Transactions on Neural Networks 9(6) (1998) 1217–1234
[22] Wickens, C. D. and Hollands, J.: In: C. Wickens and J. Hollands(eds.):Engineering Psychology and Human Performance Prentice Hall, Chapter 11 (2000) 439–479