

# A Surprise Triggered Adaptive and Reactive (STAR) Framework for Online Adaptation in Non-stationary Environments

Truong-Huy Dinh Nguyen and Tze-Yun Leong

Medical Computing Laboratory, School of Computing  
National University of Singapore  
13 Computing Drive, Singapore 117417  
{trhuy, leongty}@comp.nus.edu.sg

## Abstract

We consider the task of developing an adaptive autonomous agent that can interact with non-stationary environments. Traditional learning approaches such as Reinforcement Learning assume stationary characteristics over the course of the problem, and are therefore unable to learn the dynamically changing settings correctly. We introduce a novel adaptive framework that can detect dynamic changes due to non-stationary elements. The Surprise Triggered Adaptive and Reactive (STAR) framework is inspired by human adaptability in dealing with daily life changes. An agent adopting the STAR framework consists primarily of two components, Adapter and Reactor. The Reactor chooses suitable actions based on predictions made by a model of the environment. The Adapter observes the amount of “surprisingness” and triggers the generation of new models accordingly. Preliminary experimental results show that STAR agents are competitive in performance as compared with current approaches, while being much more cost-effective by avoiding the negative effects of historical data. Furthermore, since response and adaptability are decoupled in the framework, the adaptive component can benefit other autonomous agents in a variety of domains with non-stationary environments.

## 1. Introduction

The quality of modern games determines their entertainment values (Tozour 2002a). While recent games are able to deliver breathtakingly realistic simulations and animations, they are not so successful in demonstrating an equivalently convincing level of strategic and tactical intelligence (Schaeffer 2001). The online playing modes in most modern games aim to satisfy the customers’ demand of competing against other intelligent players. One observable difference between human gamers and Non Player Characters (NPCs) is the ability to adapt to different gaming styles. It is usually possible to beat an NPC at any level given sufficient time, and repeat the same winning strategy without the NPC turning the table around; whereas maintaining superiority over a fellow gamer is more difficult. For a game, adaptability is therefore an important

AI feature that greatly contributes to its replayability and thus its entertainment value. In real life applications, adaptability can be very useful for agents working in new or unfamiliar environments, helping to increase their vitality.

In this work, we address the problem of online adaptation, i.e., the process of self-tuning while the agent is in action. Current approaches such as Reinforcement Learning (Kaelbling, Littman, and Moore 1996) aim at solving the adaptation problem through segmenting the workflow into short episodes, during each of which there is no adaptation, and interleaving them with learning phases so as to improve the agent’s performance in the next episodes. In an episode, the environment is assumed to be stable, and all collected observation data is valid for learning. This approach performs well for settings in which the workflow segmentation can be predefined easily, such as those described in Dynamic Scripting (Spronck et al 2006) and Tactical Personality (Tan and Cheng 2008). It is not clear, however, how this approach can be applied to provide adaptability in more complex and continuous scenarios, such as a final clan battle in Massively Multiplayer Online Role-Playing Games (MMORPG). In these cases, adaptation needs to be done during the course of the battle; no clear-cut episode segmentation can be preset to ensure collected observations of strategic data in one episode are not obsolete and valid for learning.

We present here a new Reactive-Adaptive framework to facilitate online adaptability by supporting strategy adjustment using newly learnt and valid knowledge about the changing environment. The adaptation process is triggered by a surprise quantifier, hence the name Surprise-Triggered Adaptive and Reactive (STAR) Framework. Each STAR agent is assigned a “personality”, which characterizes how responsive it is to surprising events, such as how shocking the event must be to be noted or how quickly the agent changes its internal belief since the point of first suspicion.

We have evaluated our framework based on an experiment setup inspired by the Rock-Paper-Scissors game, in which the agent competes with an opponent that acts as a non-stationary environment switching from one strategy to another at unknown time points. Preliminary results show that a rational STAR agent is capable of noticing the strategic changes of its opponent, and

adjusting its belief model and actions accordingly to ensure competitive performance.

## 2. An Example Scenario

The STAR framework is targeted at a large class of problems compatible with the following typical scenario: Suppose a robot agent is given the task to move from point A to point B. Initially, the robot is given a map to plan its route. After having planned out the route, it sets off from point A. On the way to B, it encounters an obstacle. The robot must decide whether this means a thorough or partial map change that requires it to send a request to a Control Centre for map update and replan the whole route, or this is just an obstacle on the road that does not invalidate its initial plan. The act of requesting for map update and replanning may be costly. If the robot is a STAR agent, it would behave as follows. The first obstacle, being an unexpected event, arouses the agent's suspicion on a map change. Depending on the seriousness of this event, as long as the "accumulated surprisingness" has not reached the agent's surprise tolerance, it continues to move according to plan. When there have been too many surprises encountered, exceeding the agent's tolerance, it will carry out the expensive act of updating its internal planning system.

The agent in this example scenario shares the same objective as the STAR agent in many other cases, including our Rock-Paper-Scissors game: Achieving its goal while avoiding updating its internal reasoning system unless there is sufficient evidence to do so. Upon obtaining this feature, the agent is able to automatically segment its execution path into stationary episodes in a cost effective manner.

## 3. Surprise Detection and Quantification

Surprise is an important stimulus for human actions, causing between happiness and sadness to a thorough overhaul of one's viewpoint (Ekman and Friesen 2003). It has been a long going subject of research in many different disciplines, dated as far as 1872 in Darwin's thesis on his theory of evolution (Darwin 1872). Up to now, there are diverse looks on how surprise is ignited and its influence on the subject's posterior behaviors. Although the definition of surprise, i.e. what give rise to a recognizable surprising reaction and how to detect them, is still wildly debated (Kahneman and Miller, 1986; Ortony and Partridge, 1987; Meyer, Reisenzein and Schützwohl, 1997; Bartsch and Estes, 1997; Teigen and Keren, 2003; Maguire and Maguire 2009), it is generally agreed that surprise constitutes an important part of living experiences, being the driving force of maintaining the consistency of one's belief system as part of an ongoing sense making process. Based on this view, our proposed adaptive-reactive framework relies on the computational quantification of surprise as a way to denote the demand for adaptation.

### 3.1. Existing Computational Notions of Surprise

Apart from psychological views on surprise, in computational sciences, surprise notions are characterized from two major perspectives, namely objective and subjective.

Objective surprise is data-dependant and perceived the same way regardless of transferring media or perceiving entities. Information entropy, as defined by Shannon E. C. (1948), is a typical example. The entropy denotes the amount of knowledge to be transferred; the more anticipated the observed data is, the less information it carries. Therefore, it can be deduced that the less entropy a piece of data contains, the less objectively surprising it is.

Subjective surprise is, on the other hand, context- or subject-dependant, which means the same data can carry different amount of "surprisingness" to different perceiving subjects. Silberschatz and Tuzhilin (1995) mentioned one such notion of subjective surprisingness, called "subjective interestingness", to detect interesting patterns in Knowledge Discovery in Databases. In modeling visual attention, Itti and Baldi (2008) proposed a Bayesian notion of surprise, which is defined as the difference between prior and posterior distribution of model beliefs. This notion however bears a huge cost on surprise quantification at each turn of observation due to the frequent expensive model comparisons and updates. To apply it efficiently, special techniques based on domain knowledge are needed to execute these operations quickly and economically, such as using the Gamma probability density to facilitate complex visual model updates (Itti and Baldi 2008).

These existing computational models of surprise, however, are not intuitive to our daily handling of surprising events. In order to perceive surprising events, we first build up a belief system through a series of observations about related matters. Surprise is the event that challenges this system by conflicting with its predictions. The update of one's belief system should not be carried out until there is sufficient evidence that the current system is likely to be dysfunctional.

### 3.2. Our Notion of Surprise and Its Utility

In our framework, we define surprise as the discrepancy between the actual outcome and its prior prediction. Surprise is therefore a subjective quantity, which is quantified differently by different subjects maintaining different hypothesis or reasoning systems. For example, if the agent A predicts that its opponent would issue an action with probability (.3 Rock, .6 Paper, .1 Scissors), the actual outcome Paper would be less surprising to agent A than it would be to an agent B predicting (.6 Rock, .1 Paper, .3 Scissors). Also, note that the actual outcome can be expressed as a probability distribution, for instant, (0.0 Rock, 1.0 Paper, 0.0 Scissors), or (.1 Rock, .75 Paper, .15 Scissors) in the case of partial observation. In either case, surprise, being the discrepancy of predicted and observed distributions, can be calculated using a suitable metric function, such as a simple Euclidean distance function or Kullback-Leibler divergence (Kullback 1959). For

example, in our experiment, we used a modified version of Squared Euclidean Distance

$$S = \alpha(P_{pred}(a_{opp}) - 1)^2 + \sum_{a_i \neq a_{opp}} P_{pred}^2(a_i),$$

in which  $\alpha > 1$  is the emphasis coefficient,  $a_{opp}$  is the issued action and  $P_{pred}$  is the probability distribution of Opponent’s action predicted by the agent. The weight  $\alpha$  stresses on the contribution of  $a_{opp}$  to  $S$ , the individual surprisingness, making it more influential than other possible outcomes  $a_i$ .

Evaluating surprisingness contained in individual events serves as the building block of what we eventually want to detect -- systematic surprisingness resulted from the opponent’s strategic changes. This is achieved by using the **Accumulated Surprisingness**, which is the summation of consecutive event surprisingness. When the accumulated surprisingness observed in the opponent’s behaviors exceeds an upper threshold, this indicates an intrinsic change in the opponent’s strategy that would have caused the series of surprising events. These observed events serve as the evidence for the strategic change, and are used for learning the new opponent model.

At this stage, we need a mechanism to disregard surprising events caused by the effect of chance. The intuition is that surprises resulted from chance are by large followed by unsurprising events, as opposed to those resulted from a diversion of the underlying process. By discounting the accumulated surprisingness when the series of surprising events is discontinued with a predictable one, we allow unsurprising events to lessen the opportunistic surprises’ effect to a disposable value, under which the accumulated surprisingness is reset to zero. This approach enforces our agent’s immunity to random surprises.

#### 4. Surprise-Triggered Adaptive and Reactive (STAR) Framework

The STAR framework consists of two components, the Adapter and the Reactor (Figure 1). The Adapter is in charge of updating the belief model of the agent to better capture the dynamic aspects of the environment. The Reactor determines the best actions to achieve the goals. The two components are decoupled so they can be used to equip other intelligent agents with adaptability, without any principle change on the agents’ original design.

##### 4.1. Designing a STAR agent’s personality

Similar to humans, each STAR agent is surprise-avert to different levels, depending on its “personality”, which are reflected by five thresholds set at the point of creation. These values are categorized into three characteristics.

**Stubbornness** denotes the agent’s resistance to changes. It takes more evidence to convince a stubborn agent than that for a docile one. This characteristic is reflected by the

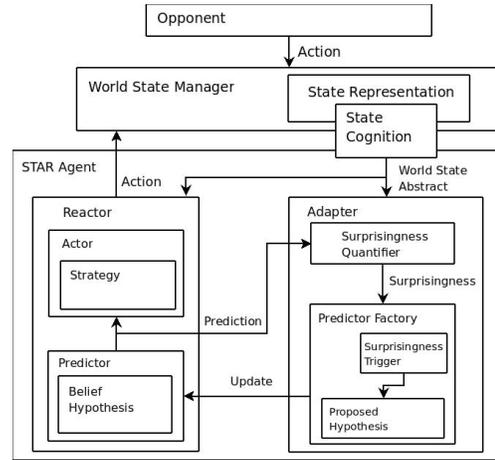


Figure 1. The Surprise-Triggered Adaptive and Reactive (STAR) Framework.

upper bound of observations’ accumulated surprisingness,  $S_{upper}$  to trigger the hypothesis proposal process.

**Hastiness** denotes the agent’s eagerness to adopt a new hypothesis, given a series of its trials. The hastier an agent is, the fewer expected trials a new hypothesis on the environment model needs to go through before it substitutes the old one. This characteristic is reflected by the number of required trials,  $N_{trials}$ , and the ratio of correct predictions over the total number of trials,  $\alpha_{succ}$ .

**Skepticism** denotes the agent’s attitude to single surprises. It is reflected through the threshold which the observation surprisingness must remain higher to be noted,  $S_{low}$ , and the discount factor applied on total surprisingness,  $\rho_{surp}$ . In other words, these values define how significant a surprise must be in order to catch the agent’s attention, and how quickly surprise decays over time.

Each STAR agent thus has different perception of and reaction to the same surprising event, even though their prior knowledge and assumptions may be the same.

In a more sophisticated game scenario such as MMORPG, a STAR personality can be used to define the adaptive behaviors of the overall strategic AI at a game level or in a gaming land. For example, an NPC, as World1, has modeled a human player’s style as being biased towards air attack. Now, suppose the player has changed his style to bias ground attack. If World1’s personality is very stubborn, or  $S_{upper}$  is very high, the human player must have used a huge army of ground based units to attack him before the NPC accumulates enough surprises to realize the switch in his opponent’s game style. This NPC is therefore inflexible in adaptation and deemed an easy target, suitable for early game levels. Deeper into the game, the NPCs can be made less stubborn, so as to make its adaptation process more sensitive to changes and quicken its reactions to the human player’s strategic switches. Other parameters defining the adaptive style allow the designer to specify a wide range of NPC’s adaptability to better challenge different players at different levels.

## 4.2. Adapter

This module encapsulates the agent’s adaptability. It consists of two subcomponents, the Surprise Quantifier and the Hypothesis Factory. The Surprise Quantifier accumulates the values of surprisingness contained in the opponent’s actions, as defined in section 2.2. Note that only actions from the current hypothesis’ nearest suspicion point (NSP) till the present account for the accumulated surprisingness. The NSP of an active hypothesis is defined as the latest time point from which the accumulated surprisingness has not been reset to zero (Figure 2).

When the accumulated surprisingness observed in the opponent’s actions surpasses the preset threshold, the Surprise Quantifier triggers the Hypothesis Factory’s

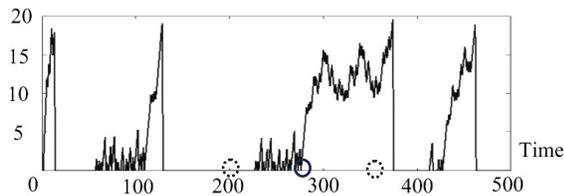


Figure 2. Total surprisingness of a STAR agent plot against time. If the agent is at time 200 (first broken circle), the current hypothesis’ nearest suspicion point (NSP) is not set, as the current hypothesis is well supporting the agent’s Predictor. If the agent is at time 350 (second broken circle), the corresponding NSP is 275 (black circle).

process to create a new hypothesis (of the opponent’s strategy) using the accumulated series of surprising data. This new hypothesis will then be matched with the new opponent actions observed, and only get adopted by the Reactor upon reaching the desired level of performance in prediction. The trials are conducted concurrently to current game flow, taking observations from the real environment to challenge the newly proposed hypothesis. The new hypothesis is given one point for correct prediction and none otherwise. After the trial period, the score acts as the performance indicator, and compared against the desired success ratio. If this new hypothesis fails, another hypothesis is generated, using more recent data. This way, the agent is able to discard unsystematic surprising outcomes.

## 4.3. Reactor

This module represents a structured guideline for individually tailored reactive agents. These agents do not follow predesigned scripted series of actions, but react according to their perceptions of the opponents’ behaviors. The Reactor consists of two subcomponents, the Actor and the Predictor (Figure 1). The Predictor maintains an internal opponent model that estimates the probability distribution on the opponent’s upcoming action at each turn. Whenever the Adapter has a well-tested new hypothesis of the opponent model, it will update the Predictor with this newly formed model so that future predictions can be made more accurately. The Actor issues

a suitable action in each cycle based on the Predictor’s suggestion and its own strategy.

For example, if the Predictor suggests that the opponent’s next action outcome would have a probability distribution of (.2 Rock, .7 Paper, .1 Scissors), and the Actor’s winning strategy is to beat the most probable predicted action, its rational choice will be Scissors. Note that this is only one possible strategy to achieve one goal namely “Beat the Opponent.” The Actor could have more than one strategy to achieve other goals.

## 5. Experiment and Analysis

As discussed in Section 3.1, each STAR agent is designed to adopt a different personality which characterizes its surprise aversion. As with humans, there is no universally good personality which guarantees top performance in all circumstances; therefore, in different settings, a suitable personality should be identified and adopted by the desired STAR agent, possibly through a training phase with sample datasets. This issue of personality training will be left as part of our future work. In this paper, we will observe the sensitivity of our STAR agent’s performance with regards to two different settings, one with high and one with low factor of randomness.

In addition, we will also compare a STAR agent’s performance with two other representative agents of current approaches in online adaptation and a baseline random player. We will analyze the results based on the agents’ performance scores and the costs taken to exhibit their respective performances.

**Rock–Paper–Scissors Game.** As a proof of concept, we created a toy game, a Rock–Paper–Scissors duplicate. In this game, the agent is to issue suitable symbols in the set {Rock, Paper, Scissors} to compete with a non-stationary opponent which changes its strategy at times unknown to the agent over an infinite number of turns.

We assume the opponent is Markovian and model opponent strategies as Non-deterministic Finite State Machines, which depict the transition probabilities from

Table 1. A sample Opponent Strategy.

Prior Action	R	P	S
Rock (R)	.2	.3	.5
Paper (P)	.4	.1	.5
Scissors (S)	.1	.6	.3

one action to the next (Table 1). We evaluate the agents using its final score against the opponent after several turns of playing, considering the fact that this is one of the most intuitive ways to judge an agent’s performance. The agent gets 1 for a win, 0 for a draw and -1 for a loss.

The temporal cost of each agent counts the total number of operations related to model construction and alteration they must execute in the course of the experiment. We argue that it is generally much cheaper to query a system than to build or modify it; therefore, operations such as outcome prediction can be neglected.

To enable easier tracking of adaptation performance, this opponent is preset to switch its strategy after every 500 turns in a total of 30000 turns spanning one simulated game’s course. This life cycle of strategies is determined to be long enough for a strategy to leave its impact on the agents’ observations, and reduce the random factor’s influence. None of the participating agents knows nor attempts to learn this characteristic of the opponent.

### 5.1. Sensitivity test of STAR agents

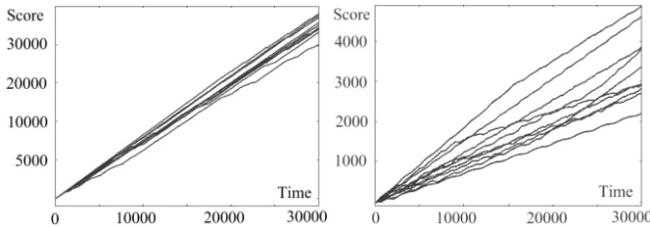
In this experiment setup, we assess one STAR agent’s performance against two different opponents with the aforementioned manner of strategy switching but containing different amounts of randomness. Specifically, Opponent 1’s strategies are less stochastic than Opponent 2’s (Table 2). The statistical data in Table 3 and 4 are collected over 100 runs of each of these two sub-experiments.

We can observe that it is easier to learn an opponent model given the same quantity of observations when

Table 2. Sample strategies adopted by Opponent (a) 1 and (b) 2.

Prior Action	R	P	S	Prior Action	R	P	S
(a) R	.05	.9	.05	(b) R	.3	.4	.3
P	.05	.05	.9	P	.3	.3	.4
S	.9	.05	.05	S	.4	.3	.3

competing against Opponent 1, the more deterministic, than Opponent 2. Our STAR agent was able to adapt



(a) Opponent 1 (b) Opponent 2

Figure 3. Charted graph of STAR’s scores in ten randomly chosen simulated runs.

timely to Opponent 1’s behaviors in a consistent way (Figure 3a). However, it is not possible to maintain this standard of performance when competing against Opponent 2. Action outcomes issued by Opponent 2 are heavily affected by random factor, confusing the learning

Table 3. Statistical data on STAR agent against Opponent 1 and 2.

Score Statistics	Min	1st Quartile	Mean	3rd Quart.	Max	Std
Opp. 1	19168	22013	22682	23556	24572	1156.5
Opp. 2	1133	3119	3743.3	4431	6129	937

process, and the agent’s score can fall as low as more than five times less than the maximal score (Table 3). This observation is reaffirmed in Table 4, when the mean cost for model building and update when challenged by Opponent 2 is eight times more expensive than that when

Table 4. Modeling cost against Opponent 1 and 2.

Cost	Min	Max	Mean	Standard Dev.
Opp. 1	1009	1682	1444.1	104.5
Opp. 2	8090	15132	11602	1156.5

challenged by Opponent 1. The STAR agent must build its hypotheses using more data and taking more attempts before finally finding the right ones.

### 5.2. Competition with other agents

The other agents involved in this performance evaluation are the random agent, naïve accumulative agent and history discounted accumulative agent.

**Random Agent.** This agent issues an action randomly in each turn; therefore, its model cost is always 0.

**Naïve Accumulative Agent.** This represents an online adaptation approach which assumes a stochastic but

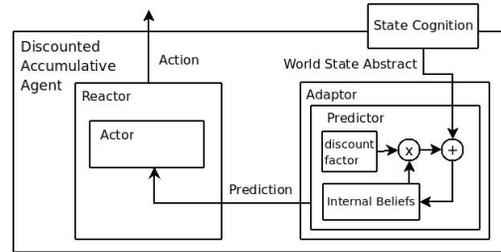


Figure 4. The structure of Discounted Accumulative Agent (DAA). The Naïve Accumulative Agent is a DAA with stationary opponent. This means observation data is never obsolete and the more observations one can obtain, the more accurate one can model the opponent’s governing process. The agent learns the opponent model by incorporating its observations into the existing model right after each turns (Figure 4). Based on outcome predictions by this model, it issues actions that beat the most probable opponent outcomes.

**Accumulative Agent with Discounted History.** This agent is a generalized version of the aforementioned naïve approach, by discounting history data so that outdated data is overshadowed by newer data in the internal opponent model (Figure 4).

One thing to note is that due to the stochastic nature of the opponent’s underlying process, a suitable discount factor must be chosen for this agent to perform adequately. An undervalued discount factor which disregards much of relevant history data would result in the loss of necessary information for learning, while an overvalued one loses its filtering effect, resulting in modeling of obsolete data. The discount factor is set to .99 in our experiment.

In order to facilitate timely adaptation, since these accumulative agents do not have any triggering mechanism for adaptation, they are allowed to fine tune their prediction model after each turn.

**Results.** As depicted in Figure 5, our STAR agent exhibits competitive performance as compared to the other agents in both cases. It is able to build up anticipation on the strategic changes of the opponent, and adapt its internal hypothesis model accordingly. Among the three other

agents, except for the random agent, which performs arbitrarily bad, the other accumulative agents are able to adaptively compete with the opponent's changes to

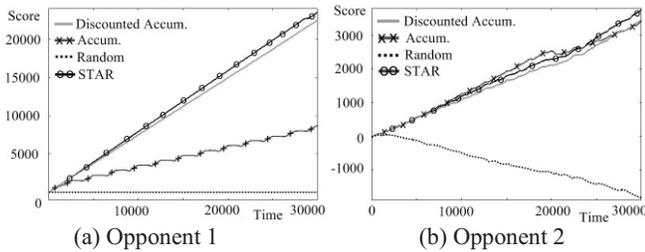


Figure 5. Performance comparison of STAR, accumulative and discounted accumulative agents, and a random agent.

different levels. While the naive approach suffers the curse of history heavily when encountering Opponent 1 (Figure 5a), and is left behind, its history-discounted duplicate and our STAR agent performed comparably well.

Table 5. Modeling cost against Opponent 1 and 2.

Cost Statistics	Random agent	Naïve Acc.	Discounted Acc.	STAR
Opp. 1	0	30000	30000	1759
Opp. 2	0	30000	30000	9247

However as mentioned above, the price accumulative agents have to pay in order to achieve good adaptive results is not always affordable. Our STAR agent requires few model operations (Table 5), especially when the random factor of the rotating strategies is not high (Opponent 1), in which it saved 94% of the total number of allowed model operations. The competing agents, on the other hand, need to update their opponent model after every turn. In complex scenarios when the operation is expensive, this approach bears a huge computational cost on the agent. Moreover, when model changes are locally confined, such expensive operations will greatly affect the agents' performance, and often leave them incapable of timely adaptation.

## 6. Conclusion

In this paper, we have presented a novel approach towards online adaptation in non-stationary environments, where the governing process dynamically changes at times unknown to interacting agents. This is a missing piece in state-of-the-art online learning approaches, such as Reinforcement Learning, which do not have any proper mechanism to detect the non-stationary elements of the learnt target. Our agent adopting the proposed STAR framework has shown to perform competitively with representative agents of existing approaches in a more cost effective manner. Even though in these initial experiments the non-stationary property is in a fairly simple form, we believe that the triggers fired by surprisingness are reasonable and inexpensive ignitions for timely model updating. Moreover, since the adaptive and reactive

components of our framework are essentially decoupled, our approach can be used as an augmentation of current learning approaches, bridging their episodes of adaptation into one continuous automatic adaptive experience.

**Acknowledgements** We would like to thank Guoliang Li for comments and suggestions on the paper. This work is partially supported by Academic Research Grants No. R252-000-309-112 and R252-000-327-112 from the Ministry of Education in Singapore.

## References

- Bartsch, K. and Estes, D. 1997. Children's and adults' everyday talk about surprise. *British Journal of Developmental Psychology*, 15, 461-475.
- Darwin, C. 1872. *The Expression of Emotion in Man and Animals*. London: J. Murray.
- Ekman, P., and Friesen, W. V. 2003. *Unmasking the Face*. Malor Books.
- Itti, L., and Baldi, P. 2008. Bayesian surprise attracts human attention. *Vision Research*.
- Kaelbling, L. P., Littman, M. L., and Moore, A. P. 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* 4:237-285.
- Kahneman, D., and Miller, D.T. 1986. Norm theory: Comparing reality to its alternatives. *Psychological Review*, 93, 136-53.
- Kullback, S. 1959. *Information theory and statistics*. New York: John Wiley and Sons.
- Maguire, P., and Maguire, R. 2009. Investigating the Difference between Surprise and Probability Judgments. In Proceedings of the Thirty-First Annual Conference of the Cognitive Science Society. Amsterdam, Netherlands.
- Meyer, W.U., Reisenzein, R., and Schützwohl, A. 1997. Towards a process analysis of emotions: The case of surprise. *Motivation and Emotion*, 21(3), 251-274.
- Ortony, A., and Partridge, D. 1987. Surprisingness and expectation failure: What's the difference? In Proceedings of the Tenth International Joint Conference on Artificial Intelligence. Milan, Italy.
- Spronck P., Ponsen M., Sprinkhuizen-Kuyper I., and Postma E., 2006. Adaptive Game AI with Dynamic Scripting. *Netherlands: Springer Netherlands*.
- Schaeffer, J. 2001. A gamut of games. *AI Magazine* 22(3).
- Shannon, C. E. 1948. A mathematical theory of communication. *Bell system technical journal* 27.
- Silberschatz, A., and Tuzhilin, A. 1995. On subjective measures of interestingness in knowledge discovery. In *Proceedings of the 1st international conference on Knowledge Discovery and Data mining (KDD '95)*, 275-281. AAAI Press.
- Tan, C. T., and Cheng, H. L. 2008. A combined tactical and strategic hierarchical learning framework in multi-agent games. In *Sandbox '08: Proceedings of the 2008 ACM SIGGRAPH symposium on Video games*, 115-122. New York, NY, USA: ACM.
- Teigen, K.H., and Keren, G. 2003. Surprises: Low probabilities or high contrasts? *Cognition* 87: 55-71.
- Tozour, P. 2002a. The evolution of game AI. *AI Game Programming Wisdom* 3-15.