Accepted Manuscript

A complex event processing framework for an adaptive language learning system

Dawei Jin, Si Shi, Yin Zhang, Haider Abbas, Guolin Pan

PII: S0167-739X(17)31281-5
DOI: https://doi.org/10.1016/j.future.2017.12.032
Reference: FUTURE 3870

To appear in: *Future Generation Computer Systems*

Received date : 16 June 2017
Revised date : 16 November 2017
Accepted date : 22 December 2017

A Complex Event Processing Framework for an Adaptive Language Learning System

Dawei Jin[a,*], Si Shi[b], Yin Zhang[a], Haider Abbas[c,d], Guolin Pan[e]

[a]*School of Information and Safety Engineering, Zhongnan University of Economics and Law, China*
[b]*School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China*
[c]*King Saud University, Riyadh 11451, Saudi Arabia*
[d]*National University of Sciences and Technology, Islamabad 44000, Pakistan*
[e]*School of Economics and Management, China University of Geosciences, Wuhan, China*

Abstract

Ubiquitous learning applications and worldwide educational websites such as MOOC (Massive Open Online Courses) are rapidly producing large volume of user data. Current delayed analysis processing in adaptive language learning systems is difficult to cope with the high-speed and high-volume data streams. To overcome this problem, we introduce a complex event processing (CEP) framework for an Adaptive Language Learning System. The system consists of an event adapter sub-system that can process various inputs such as voice, video, text and other interaction events. The event adapter extracts relevant data to support the operational events module, the learning activity events module and the learner knowledge space events module. These three modules in the event hierarchies provide support to the learner adaptation and learner visual analytics modules. In this study, we conduct three simulations to evaluate the initialization time, delay time and throughput of the proposed system. Each of the experiments simulates 1000 learners and 1000 rules and generates 10 events per second. The results indicate the CEP framework is efficient with a

*Corresponding author
Email addresses: jdw@zuel.edu.cn (Dawei Jin), shisi_cc@126.com (Si Shi), yin.zhang.cn@ieee.org (Yin Zhang), hsiddiqui@ksu.edu.sa;haiderabbas-mcs@nust.edu.pk (Haider Abbas), panguolin@163.com (Guolin Pan)

processing delay of less than 1.2 μs and throughput of 80,000 events per second. We conclude by discussing the study's implications and suggest ideas for future research.

Keywords: Big Data, Adaptive Educational System, Language Learning, Complex Event Processing, Stream Processing

1. Introduction

Adaptive Educational Systems (AES) have been popular in recent years. Compared with Learning Management Systems (LMS), which offer limited adaptivity and personalization, adaptive educational systems use intelligence algorithms to adapt to student's learning style, enhance learning performance, accelerate goal achievement, reduce navigational overheads, and improve overall student satisfaction [1]. Research about adaptive, reactive, rich internet applications makes it possible for educational systems to adapt based on current user's context and actions [2]. Lately, several new trends in technology and learning are influencing the future development of adaptive educational systems. First, learning is already ubiquitous due to the increasing use of mobile smart devices in the education domain. Learning applications on different types of learning devices such as notebooks, smart-phones and other mobile devices provide learners with a more convenient way to study [3]. Adaptive learning systems must take into consideration these different types of devices and applications. Second, with the development of Massive Open Online Courses (MOOC) more and more educational platforms are being used globally [4]. The merging of educational systems into cloud-based educational systems with distributed service centers, data centers and resource centers is inevitable. Adaptive educational systems must consider their role in the new form of worldwide educational systems. Third, big data produced by ubiquitous learning devices and worldwide educational test and assessment systems generate information that is potentially useful for improving education 'delivery' [5]. Instead of producing coarse analysis, big data analytics from devices, sensors and internet-of-things provide opportunities for adaptive systems to track and process fine-grain records of learners to produce a comprehensive model of learners' behavior.

However, current big data analysis technologies, such as reinforcement learning [6], remain limited to specific intellectual areas and fail to cope with every phenomenon in real world [7].Complex Event Processing (CEP) provides an effective potential solution to cope with different types of learning data in real time. As a stream processing technology, it is designed to handle data volume, data variety and data velocity and output digested data in real time to all requesting units. The architecture of the complex event-processing system is rather comprehensive [8, 9]. It consists of several models such as functional, processing, deployment, interaction, data, time, rule and language. There are already existing CEP systems such as Cayuga [10] and SASE [11, 12] and commercial software such as Esper by EsperTech, BusinessEvents and Stream Base by TIBCO and the Sybase CEP system. Table 1 demonstrates CEP applications in various domains. In the education domain, complex event processing recently appears in learner models to exploit large-scale cognitive modeling [13].

Despite these recent developments, researches in complex event processing have remained isolated from research on education systems. To our knowledge, complex event processing technologies have not been applied in educational systems, let alone language learning. There are several issue that should be considered in applying complex event processing technique to adaptive language learning domain. First, complex event processing pursues extremely low latency at the cost of the ability to perform complex analysis, which is necessary for analysing learner's current situation and the according reaction. Second, one-pass stream-based processing model of complex event processing differs from traditional storage-based model. Analysis must perform without keeping any tracks of passed samples in learner analysis and display filtering. Last but not least, instead of considering each data separately from the others, complex event processing considers complex event patterns that involve the occurrence of multiple, related events. However, there is no universal rules to organize events in complex event processing yet.

Table 1: Applications of complex event processing in various domains.

Author	**Time**	**Context**
Yao et al. [14]	2011	Medical treatment in smart hospitals
Terroso-Saenz et al. [15]	2012	Traffic congestion detection
Xiao et al. [16]	2012	Power management in wireless data communication
Cheng et al. [17]	2012	Risky event detection for mining safety
Douglass [13]	2013	Large-scale cognitive modeling
Stipkovic et al. [18]	2013	Pervasive computing
Chen and Chen [19]	2014	Intrusion detection for network security
Qiao et al. [20]	2014	District heating service
Jayan and Rajan [21]	2014	Intrusion detection (log data processing) for network security
Llinas [22]	2014	Multi-dimensional sonification processing
Terroso-Sáenz et al. [23]	2015	Perceive the contextual information of a vehicle
Choi et al. [24]	2016	NASA's Deep Space Network (DSN)
Costantini et al. [25]	2017	Cognitive Robotics

To solve these problems, in this paper, we present an adaptive language learning system using complex event processing to handle events produced by ubiquitous learning devices and worldwide educational systems. In order to speed up large-scale high-volume adaptation, we assume the adaptation rules have been predefined by experts, and our framework focuses on speed up learner state tracking in runtime instead of researching the adaptation strategy.

We organize the paper as follows. First, we discuss the trends of adaptive educational systems and introduce complex event processing. Next, we introduce the concept of event, rule model and discuss how to design an adaptive language learning system using an event model and language model. Next, we present the event hierarchies in the adaptive language learning system including three levels: operational events, learning activity events and knowledge space events. Next, we describe the framework and workflow of the adaptive language learning system. Finally, experiment results of the adaptive system and its limitations and suggestions for future work are discussed.

2. Foundations of Complex Event Processing

There are some concepts with specific definition in universal complex event processing. In this section, we give an introduction of these concepts and apply these concepts to adaptive language learning domain.

2.1. Event

An event is an object that represents or records an activity that happens, or is thought of as happening [26]. A simple event is denoted as $E_s =< id, time, type, attr >$ where id denotes the unique identifier of the event, $time =< t_0, t_1 >$ denotes the event occurrence time consisting of the begin timestamp t_0 and end timestamp t_1 (begin and end timestamps of a simple event are usually the same), type denotes the event type and $attr =< attr_1, K, attr_n >$ denotes the set of event attributes. A complex event is denoted as $E_c =< id, time, type, attr, memb >$ where $memb =< e_1, K, e_m >$ denotes the member events that constitute the complex event. Simple events are atomic and instantaneous events that cannot be sub-divided. Complex events are abstracted from their sub-events and represent their member events. For example, operations such as clicking and typing in adaptive educational systems are simple events because they are directly produced by the system and occur instantaneously. Learning a lesson is a complex event because it cannot be obtained from the system. Only when various events such as interactive operations in the user interface, listening to a dialog, writing essays, and other activities are collected and extracted can a system infer that the learner is learning a lesson.

2.2. Event constructors

Event constructors indicate the relationship between events. A CEP engine will filter events that cannot satisfy the relationship the constructors indicate during runtime. Although there are no uniform Event Processing Language (EPL) standards like SQL in the relational database domain, there are still event constructors that a CEP engine is able to support. Table 2 shows the constructors supported in different commercial CEP systems [9, 27].

1. *Selection (S).* Selects the events whose attributes satisfy the constraints that other constructors indicate.
2. *Projection (P).* A mapping function, where the attribute values of an event can be obtained from the attribute values of its sub-events.
3. *Disjunction ($\vee$).* Represents either of two events occurring.
4. *Conjunction ($\wedge$).* Represents both of two events occurring. Conjunction can be obtained by combining sequence and disjunction.
5. *Negation ($\urcorner$).* The event should not occur at all.
6. *Sequence (;).* The events occurring one after the other in time.
7. *Iteration ($*$).* An event occurring several times in sequence.
8. *Windowing (W).* Restricting events should occur within a specified time window or length window. The window can move (sliding window) or jump (batch window) over event streams.

Table 2: Universal event constructors supported by different commercial CEP systems.

CEP Systems	**Esper**	**Tibco BE**	**Stream-Base**	**Aleri SP**	**IBM System S**
Selection (S)	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
Projection (P)	$\surd$		$\surd$	$\surd$	$\surd$
Disjunction ($\vee$)	$\surd$	$\surd$	$\surd$	$\surd$	
Conjunction ($\wedge$)	$\surd$	$\surd$	$\surd$	$\surd$	
Negation ($\neg$)	$\surd$	$\surd$	$\surd$	$\surd$	
Sequence (;)	$\surd$	$\surd$	$\surd$	$\surd$	
Iteration ($*$)	$\surd$	$\surd$			
Windowing (W)	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
Hierarchies (H)	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$

9. *Hierarchies (H).* A set of events level and a set of rules for computing events at each level as abstractions of patterns of events from the levels below.

Through combinations of different constructors, rule managers are able to describe the event abstraction rules and the CEP engine will capture event patterns in the event cloud according to the event abstraction rules.

2.3. Event abstraction rules

Event abstraction rules detect event patterns satisfying the constraints that event constructors indicate. Instead of being triggered by a single event, event abstraction rules in complex event processing are triggered by event patterns, which make it possible to process various kinds of events in a rule. The language used to describe event abstraction rules is the Event Processing Language (EPL). With event processing language, users can submit a query statement to the CEP engine. Then the CEP engine will compile the statements and detect the event patterns automatically. In our work, the CEP framework uses

Table 3: Universal event constructors supported by different commercial CEP systems.

Situation	**Rule Definition**	**Adaptive Actions**
S1: Detecting whether the student is a visual learner or not.	insert into VisualActivity-Summary selectWebEvent.learnerID, count (*) as sum from WebEvent.win:time(60 min) where WebEvent.visulType = image	If the student tends to browse images more frequently, and present more visual information instead of text.
S2: Student keeps studying for a long time and does not take a break.	select * from pattern [every (timer:interval(3 hour) and not RestEvent)]	Show a warning to remind the learner to take a break.
S3: The performance of a student is usually good and stable but suddenly deteriorates.	select A.learnerID, A.topicfromTestRsult match_recognize (pattern (A+B) define A as A.score¿= 60, B as B.score¡= 45)	The system can present more complementary materials to improve the learning performance or lead to other related courses according to the weakness.

the adaptive language learning system Esperin, which is an open source CEP system widely used in many domains.

As for the adaptive language learning system, the event abstraction rules are designed according to various adaptive situations. Examples of detecting situations are shown in Table 3 with event abstraction rules and corresponding adaptive actions [28]. S1 is the aggregation query example used to detect visual learners. S2 gives the example of an event pattern to detect a tired student. S3 shows the match recognized using the familiar syntax of regular expressions to detect the sudden decline in performance.

Event hierarchies denote the set of events and event rules describing the extract relationship between events. Event hierarchies extract data streams to support information flow which organizes event abstractions into levels [26] consisting of two elements: (1) a sequence of levels of activities and associated event types, and (2) a set of event abstraction rules indicating how high-level events are abstracted from low-level events.

3. Design Issues

3.1. From complex event processing to language learning

In adaptive language learning systems, a complex event processing engine can capture real-time precise learning activities and voice information to allow adaptive fine-tuning of the learning sequence. The design process of adaptive language learning systems involves three key steps:

1. Design the event hierarchies of language learning. The key in using CEP is to define all information in the application system at the appropriate and clear event hierarchies. Because an event hierarchy defines a set of levels of activities and a set of rules for computing events at each level as abstractions of patterns of events from the levels below, the first step is to design event types in each level. Only when the structure of the event hierarchies has been decided can the rules be designed.
2. Design the event abstraction rules of learning activities. Event abstraction rules are generated by event processing language with a combination of operators. In adaptive language learning systems, the rule managers are able to personalize their own learning event patterns and realize real-time learner adaptation.
3. Design the adaptation process. The adaptation process is vital in adaptive educational systems. Current approach utilized learning styles that provide input for the adaptation process focus on static learner modeling such as age and gender instead of dynamic learning behavior [29, 30, 31]. The complex event processing approach makes it possible for systems to make adaptations according to dynamic learning behavior.

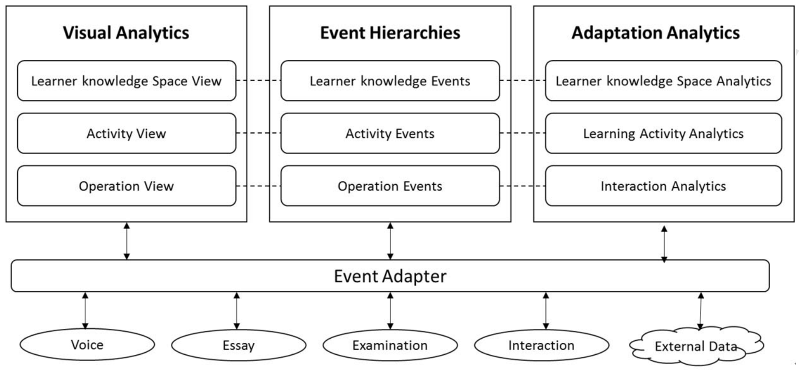

Figure 1: Event hierarchies of adaptive language learning.

3.2. Event hierarchies in language learning

Event hierarchies represent the semantic data flow in a complex event processing framework. As has been previously mentioned, an event hierarchy defines a set of activities at different levels and a set of rules for computing events at each level starting from the lower-level events. Rule managers or experts manage the abstraction rules which are stored in the rule database. The CEP engine compiles the rules and signal critical situations. Event hierarchy definitions are flexible and complex, so that only experienced domain experts can create rules and make adjustments accordingly. In this paper we introduce a reference framework for an adaptive language learning system (shown in Fig. 1) instead of a static framework because language learning systems should be allowed to change according to the context.

Event adapters transform raw data from the event producer into simple events at the lowest level. In an adaptive language learning system raw data includes voice data from speech processing devices, test data from online educational systems and examination systems, interactive data from hypermedia applications and other data sources necessary for evaluating a learner's learning state. Besides online data, archived learner profiles data from databases or files is also an important basis for learner knowledge space evaluation.

One of the most challenging problems in language learning systems is that only certain raw data can be transformed directly into simple events such as web interactions. Most of the raw data, especially speech data from speech processing devices, which plays a vital role in language learning and is an important reference for the learning state, cannot be transformed into simple events directly. The event adapter then pre-processes raw data into events that can be further processed by a complex event processing engine. In addition to the adapter interface, the system can process other data sources not included in the system.

Simple events are basic events at the lowest level, directly transformed from raw data, and have not been aggregated. A single simple event cannot provide complete information on a learner's knowledge space. Only when a large number of events of different types from different data sources have been analyzed can learner knowledge space be established. In an adaptive language learning system, simple events are usually operational events or machine data such as speech data, test data and operations in applications.

At a higher level, which we might call the learning activity level, some events might be defined such as pronunciation errors, skipping a training test, learning a course, pausing learning activities for a predetermined time or other relevant learning activities from patterns of events at the operations level. Events at this level might not reflect the complete learner's learning state but do contain actual meanings. According to the detected events, the system can give a warning, provide recommendations for improvement or run an adaptation process according to the learning activities.

The complex events at higher levels are abstract and contain more semantic meanings. A higher level in an adaptive language learning system might be the learner knowledge space view which combines the events at learning activity level to provide a reference. An adaptation module can use the events at learner knowledge space. For example, if a learner is interacting rather than learning alone, he/she may be an active learner and the system may adjust the learning sequence to fit the learner's learning style and cognitive ability. Cognitive ability can be inferred from Intelligence evaluation technologies to assess intelligence [32].

Once the complex events have been detected, the events will be transformed by the event adapter and transferred to the corresponding event consumer, which is the adaptation analyzer module in an adaptive language learning system.

3.3. Event hierarchies implementation based on event processing network

The key to speeding up learner state tracking in runtime is to perform effective and efficient adaptive language learning event hierarchy implementation. In complex event processing, event hierarchies are implemented with event processing network (EPN). An event processing network consists of four components: event producer, event consumer, event processing agent (EPA), and event channel. An event producer represents an event stream source of the EPN. An event consumer represents an event sink of the EPN. An event processing agent is responsible to enrich events, perform pattern matching, derive new events, and publish them to an event channel. An event channel receives events produced by event producer or event processing agent and delivers them to event consumer or event processing agent. As processing component and delivering component, performance of event processing agent and event channel is vital for system performance.

In this section, we propose a complex event detection algorithm working in an event processing agent for adaptive language learning.

At the beginning, run set in event processing agent is an empty set. When event channel keep delevering events, agent will search for all the run in its run set (Line 4). If the event does not beyond the time window of the run and it satisfies the predecate of the run, the event will be buffered and the run's state will transfer to the next state (Line 6-7). If the run has arrived at accept state, agent will create a complex event and publish the event to the event channel (Line 9). As the run has finished, it will be deleted from run set (Line 10). Then system will test if the event can start a new run by evaluating whether it satisfies the predicate (Line 17). If so, a new run is created and added to run

Algorithm 1: Complex Event Detection by EPA

```
Initiate run set R ← ∅;
while stream from event channel not empty do
    e ←Latest event in stream;
    for r ∈ R do
        if checkTimeWindow(e,r)=Ture ∧ evaluatePredicate(e,r)=Ture
        then
            r.buff ← r.buff + e;
            r.state ← r.state.next;
            if r.state is accept state then
                create a (complex) event e_c and publish to event channel;
                R ← R − r;
            end
        end
    end
    create r;
    r.buff ← ∅;
    r.state ← firststate;
    if evaluatePredicate(e,r)=Ture then
        r.buff ← r.buff + e;
        r.state ← r.state.next;
        R ← R + r;
    end
end
```

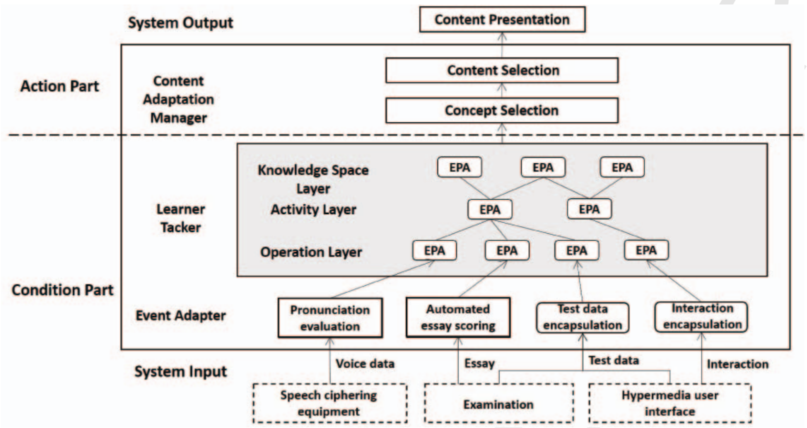

Figure 2: Framework of an Adaptive Language Learning System.

set.

4. Framework of Adaptive Language Learning System (ALLS)

4.1. Architecture overview

As mentioned earlier, the behavior tracker is the core of the adaptive language learning system. Fig. 2 shows the modularized adaptive language learning system. Each modules is responsible for processing a specific step in the adaptive sequence.

Data from language learning activities such as practicing speech, taking an examination, listening a conversation, and so on, in platform, will be accepted as input of adaptive language learning system. Adaptation trigged by learning activities consist of two parts - condition part and action part. Condition part accepts system input, transfers them into simple events, and tracks the learner's state. Action part determines the concepts and contents present to learners according to the learner' state information from condition part.

1) Condition part. While test data and interaction data can be encapsulated as simple events naturally, voice data from speech ciphering equipment and written essay from examination cannot be processed directly by complex event

processing engine. Event adapter provides transformation from unstructured data such as text and audio, to events with simple or nested attributes. Voice event adapter transforms speech by evaluate the pronunciation and essay event adapter transforms essay via automated essay scoring technology. Events from event adapters are reflected as the lowest layer events in event hierarchies and will be processed by event processing network (EPN) in learner tracker. Event processing agents (EPAs) at operation layer identifies learner operations like listening, speaking, doing examination from simple events and submit them to activity EPAs. Activity EPAs accepts learner operations to identify the current learning activity. Learning activities are then abstracted to current learning state of learner by knowledge space EPAs.

2) Action part. Once learner tracker has identify the current learning state of learner, content adaptation manager select the concept based on learner knowledge space and then select the content that will be present to learner. Finally the selected content will be displayed to enhance student learning performance.

4.2. System input and event adapter

1. **Evaluation of pronunciation quality**: As an adaptive language learning system, one of the most important inputs is the voice data of learners. However, the system takes evaluations of pronunciation quality as input instead of direct voice data. The reason is that raw voice data cannot provide information about a learner's learning state directly. Communication modes based on speech have been studied to provide Internet speech information services [33], which makes it possible for an adaptive language learning system to make use of speech interactions. From the evaluation of pronunciation quality, a system captures the real-time change in the learner knowledge space and makes adjustments. Speech recognition technology plays an important role in Computer-assisted Language Learning (CALL) [34]. Currently pronunciation evaluation in CALL is mostly based on speech recognition. Because of the important role in speech recognition, the Hidden Markov Model (HMM) is widely used in

evaluation of pronunciation quality [35, 36]. Pronunciation quality evaluation based on HMM uses trained HMM to compute the score of voice data. Then the result of that evaluation will be the input of the Behavior Tracker Module to estimate the learner's knowledge space.

2. **Automated essay scoring (AES)**: As an important method to evaluate the writing skills of language learners, automated essay scoring is also a vital input of an adaptive language learning system. Currently most AES is considered to be a supervised learning issue. Measurable features reflecting essay quality are extracted and scored using different models such as regression [37, 38, 39], classification [40, 41] and ranking [42]. The features used in automated essay scoring include non-text features such as the number of words, length of words and sentence structure and text features such as lexical and part-of-speech features, syntactic features and topic features.

3. **Interactive operations**: Detecting the pattern of interactive operations is especially suitable for evaluating the current learning state and very difficult to use in traditional language learning systems because the operations are too minor to be captured and processed. Complex event processing is thus appropriate for handling interactive operations due to its capability of processing different types of data streams in real time. Clicking, moving, typing and other interactive operations in adaptive language learning systems are normal and seem meaningless. However, such actions may indicate emotional learning states like inattention (randomly clicking), perfunctoriness (choosing certain answers in choice questions), and fatigue (learning for a long time). The system can react immediately or as an input to the adaptation process.

4. **Test scores**: Test scores are widely used in educational systems and examination systems. Test scoring with complex event processing helps to find problems in real time and gives advice or corrections. Although the final score is important, the score for each question will also express the knowledge space of the learner. For example, a learner may do badly with

certain words providing additional teaching or testing about the words promptly can help the learner to master these words. Otherwise, once the learner's established habit of misperceptions they will take a long time to correct.

5. **Other data sources**: Because an adaptive language learning system is extensible, other data sources that can benefit the evaluation of a learner's learning knowledge space can be imported to the system via an event adapter interface. How the imported data is transformed to simple events should be defined in the event adapter interface. Once the interface is implemented, the data from that kind of source can be used in the system like built-in data sources attribute to CEP's capability of processing various kinds of data.

4.3. Behavior tracker

Behavior Tracker follows the tracks of learners' learning activities at all times and provides the foundation of the adaptation process. The Behavior Tracker in an adaptive language learning system uses complex event processing to handle different types of learner knowledge space evaluation. The effectiveness of complex event processing guarantees timely corrections to learners' learning behavior and reflection of their knowledge space. There are two main modules of a behavior tracker: the event hierarchy base and runtime tracker. Rule managers rearrange the event hierarchies and the runtime tracker is responsible for tracing the learner's behavior when they are using the system. The event hierarchy base stores the event hierarchies of the system, including a sequence of event levels and associated event types, and a set of event abstraction rules for each level. The information about event hierarchies will be stored in the database and can be edited at any time. Before learners use the system, the event processing language compiler will parse each rule statement and construct a detecting model containing the constraints the statement describes. When running the system the runtime tracker will follow the trail of the learner and keep estimating whether the activities of the learner satisfy the detecting model. Once a complex event has been detected it will be sent to a certain event consumer in the adaptation analyzer module and run the adaptation process.

4.4. Content adaptation manager

According to the current knowledge space of learner captured by learner behavior tracker, content adaptation manager selects the learning concepts based on user model and domain model. Then content adaptation manager selects which content from media space to be presented. The rules in content adaptation manager is predefined by content experts.

4.5. System output

Learner behavior tracker based on complex event processing continuously captures current knowledge space of learner and content adaptation manager determines the presented content. Finally system adjusts its content into the content selected by content adaptation manager.

5. Runtime Workflow

The runtime workflow of the adaptive language learning system has four steps (shown in Fig. 3): acquiring data, tracking behavior, adapting learning activities and presenting educational content.

5.1. Acquiring data

As introduced above, the ALLS system will collect various kinds of data to evaluate the learner's knowledge space and provide the basis for the adaptation process. Data such as speech data, test data, interactive operations, and other data sources necessary for evaluating a learner's learning state will be transformed to simple event form by event adapters and be sent to the behavior tracker. The input of the system is extensible as long as the adapter interface has been implemented.

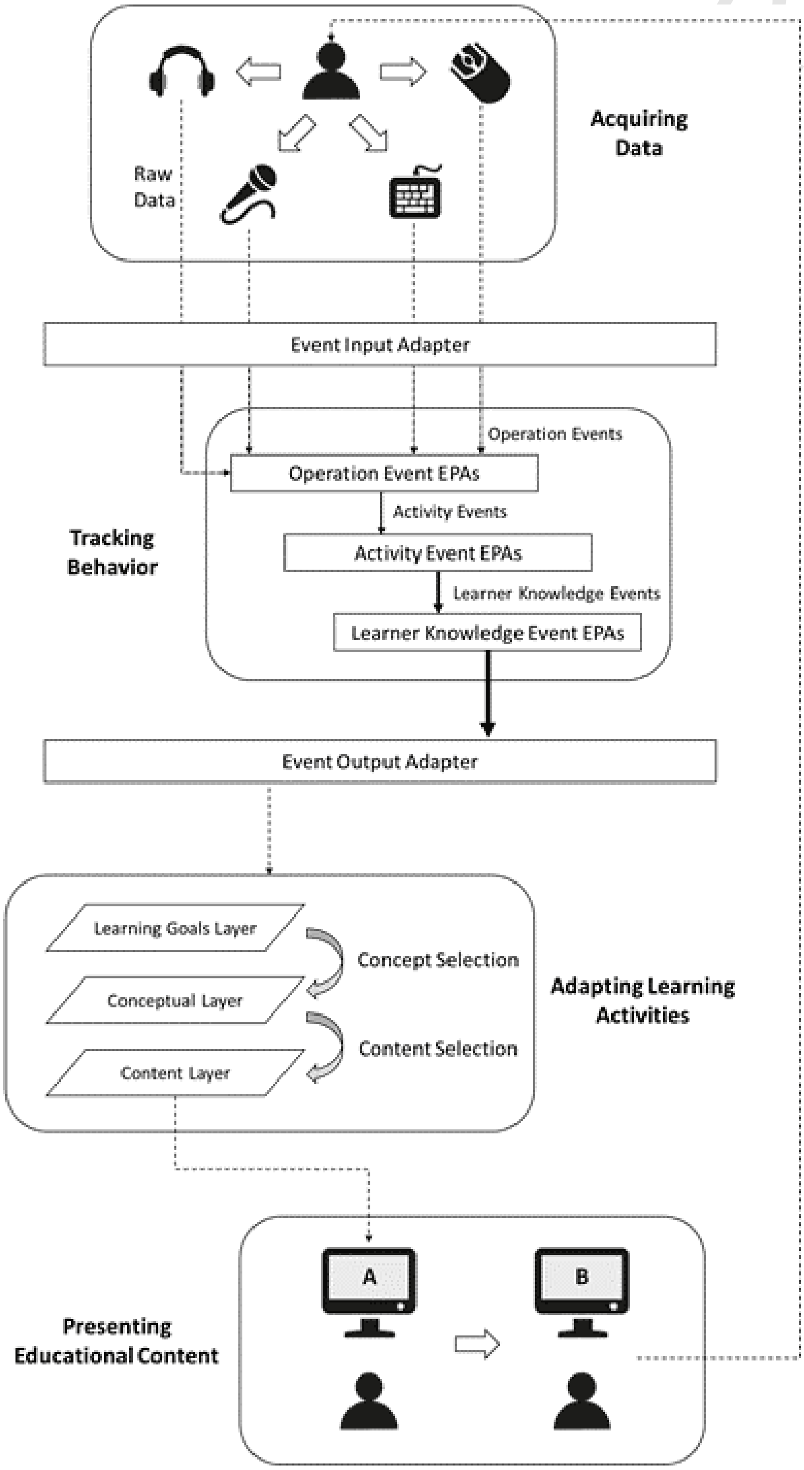

Figure 3: Runtime workflow of the adaptive language learning system.

5.2. Tracking behavior

Behavior tracking is one of the most important modules of the adaptive language learning system. It is designed with complex event processing and powerful data processing capability. Once the event hierarchies of the system have been established, the behavior tracker will run the detecting process continuously unless the statement has been detected. The detected events will be published according to the event consumer in the adaptation analyzer module as a basis for the adaptation process. Because of the complexity of adaptation, the behavior tracker should be able to process hundreds of rules at a time. Therefore, the processing capacity of the complex event-processing engine supporting the system will be the most vital index to evaluate the performance of the system.

5.3. Adapting learning activities

Learning activity adaptation is vital in adaptive educational systems. In ALLS the learning concepts will be selected based on the learner knowledge space that comes from the behavior tracker and content is selected based on characteristics and preferences [43]. The adaptation process is complex and ALLS' goal is to increase its effectiveness.

5.4. Presenting educational content

Educational content presentation is the last stage of the runtime workflow cycle. It influences the effectiveness of the system's intervention. The content presenter is the user interface of the adaptive language learning system and designed to be intuitive and understandable. Learners will be adapted to and corrected seamlessly.

6. Simulation

6.1. Experimental data and evaluation standards

The goal of the CEP system is to accelerate and fine-tuning the learner's learning state tracking in language learning in order to handle the data volume

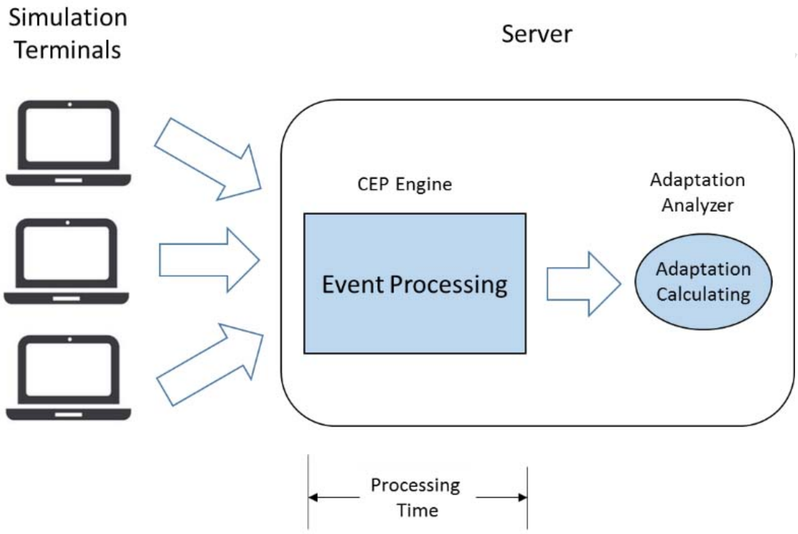

Figure 4: Architecture of the evaluation setup.

and velocity from numerous mobile devices and worldwide educational test systems. Therefore, we evaluate the performance of the behavior tracker module to ascertain if the adaptive system can support the volume and velocity of learning events.

We construct an experiment setup (shown in Fig. 4) for evaluating the proposed system. Simulated terminals were used in the experiments to represent actual learners in order to control the number of events they created [44, 45]. We evaluated the processing time in the server to avoid the influence of network delays.

The test system used a x64-based PC with Intel(R) Core(TM) i5-3470 CPU and 8,080MB of RAM running a Windows 7 Ultimate operating system with Java version 1.8.0_60 and Esper version 5.1.0.

Achieving low delays and high throughput in different situations are critical to the success of ALLS. There are three major challenges that prevents large-scale high-volume real-time adaptive language learning platform run smoothly. First, with time goes by, system may breakdown due to block resulted from continuously coming data stream. Whether system can keep running smoothly and steadily overtime is vital for large-scale high-volume adaptive language learning platform. Second, when the number of learners becomes larger, system has to handle more data at the same time and becomes more likely to face long latency. Third, our system does not limit the number of learner state tracking rules. Complete and accurate learner knowledge space evaluation usually needs more tracking rules to capture learner state. However, more rules leads to more computation time, which may result in long latency or even breakdown. We designed experiments to evaluate the performance of the proposed system from three aspects: performance over time, performance under an increasing number of learners and performance under increasing rules.

6.2. Experiment 1: performance over time

In Experiment 1 the availability of the system over time was evaluated to check whether the system can keep working at a steady state. 1000 simulated terminals sent 10,000 events to the server running 1000 rules. Fig. 5 shows the percentage of events with different processing times over time. As is shown in the figure, the percentage of events with processing times less than 5 μs increased from 93% to 96% and then slightly wobbled around 96%, which means overall the processing time of events was reduced at first and then became stable. At the very start, the system does some preparatory work, which will influence the process of events. After the preparatory work has been completed, the system processes events at a steady state where about 96% events can be processed in less than 5 μs. As time goes on the processing time does not change rapidly.

From the result we can conclude that the system will not breakdown in the simulation environment because events can be processed in a quite short time and will not lead to block in system. It is because complex event processing consider user learning data as data stream and use several techniques to speed up stream process such as continuous queries, one-pass learning, incremental computing, and so on. Current educational systems are unable to produce real-time adaptation on the huge amount of learning data streams because the underlying

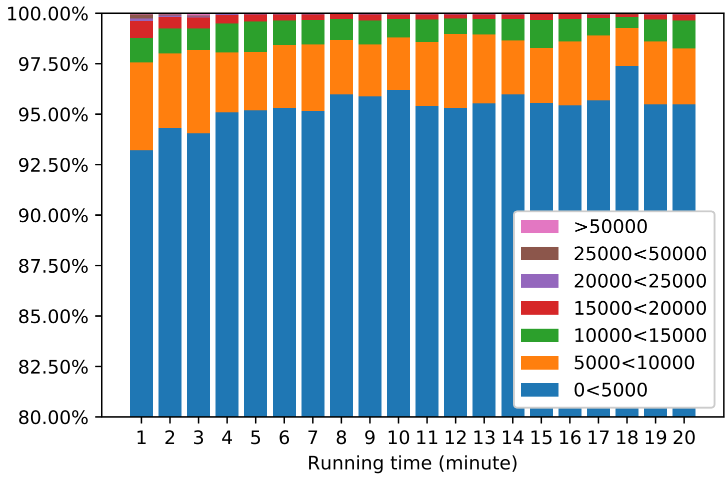

Figure 5: Percentage of events with different processing times (unit: ns) over time.

data processing method encompassing a centralized collector, suitable to store and process data in batch but are not adequate to process a variety of large continuous data streams [46].

6.3. Experiment 2: load test - number of learners

In Experiment 2 we studied how the engine delay changes with the increase of learners to find out the capacity of the proposed system. In the experiment, the server ran 1000 rules and each terminal sent 10 events per second to the server.

Fig. 6 shows the changes of average engine delays per event with different numbers of terminals over time. As can be seen in the figure, the increase of simulation terminals leads to an increase of average engine latency per event. The reason is that the more terminals, the more events to be processed, and the more processing time. However, even when the processing time becomes longer, the maximum value stays less than 1.2 μs, which is still too short a time to be noticed.

With increase of terminals, computation time of each event becomes longer

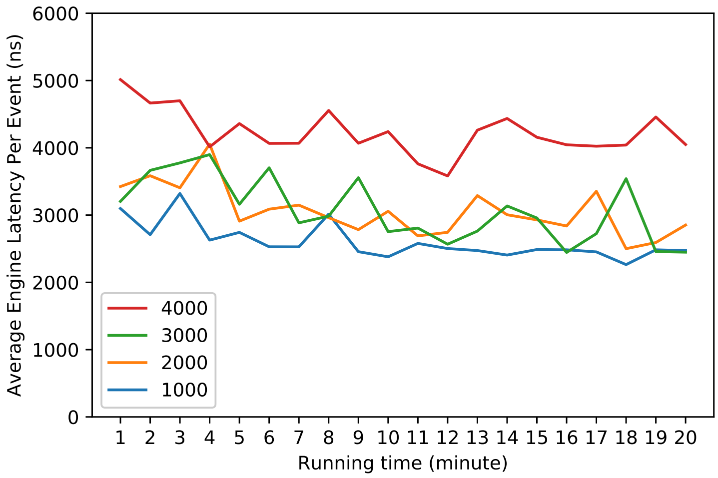

Figure 6: Average engine delays per event with different numbers of terminals over time.

due to larger number of events to be processed. The experiment illustrates that even with 4000 learner studying at the same time this framework can work smoothly and can display dynamic learning content in real time.

6.4. Experiment 3: load test - number of rules

In Experiment 3 the throughput of the system, running different numbers of rules was evaluated to determine how the increase of rules influences the system performance. There were 1000 simulation terminals and each terminal sent 10 events per second. Fig. 7 shows the throughput of different numbers of rules running in the system. Throughput here is the total number of events processed divided by event processing time, not the number of events processed per second. The trend of throughput is similar when the number of rules increases. The throughput increases unsteadily at first and becomes steady as time goes on. At a steady state the system can process about 80,000 events per second, which is quite large for language learning systems.

More rules does not influence the throughput of our system. The reason is that in complex event processing, rules that are not satisfied by events will be

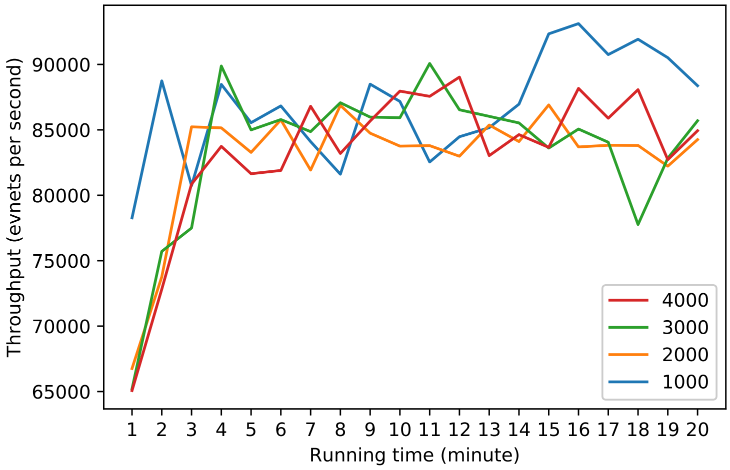

Figure 7: System throughput when the numbers of rules increase.

skipped immediately. The result allows teachers and experts to design more rules to track the knowledge space of learner.

7. Conclusions

Adaptive educational systems can expect ubiquitous learning devices and worldwide educational test and assessment systems to churn out larger volume of educational data. Traditional batch processing based on a relational database will be challenged to perform real-time adaptation. This paper presented a novel approach using complex event processing as a behavior tracker to process learning interaction events and provide real-time fine-tuning in an adaptive language learning system. It is a pioneering work that CEP is adopted in language learning and adaptive educational systems. This approach helps to accelerate the learner's learning state tracking in language learning and provides a way to improve the adaptive fine-tuning process. The performance evaluation in terms of throughput showed that our approach is feasible and effective.

Although the proposed system is capable of processing large volume of stream data in real time, we need to bear in mind its limitations. First, because complex event processing is designed for massive amounts of data flows, it is not effective for processing archived or small amounts of data. Therefore, it is more suitable to complex dynamic language classrooms [47] or massive online learning [48] instead of standalone educational application without interaction. Second, complex event processing is not suitable for direct knowledge discovery process from data such as the key factors influencing learning performance. Data mining technologies should be used to support and complement CEP activities [49]. Third, due to the demand for high processing speed, data in complex event processing is processed in memory without redundancy; a data recovery plan should be in place in the event of failure.

The proposed approach has three implications for the design of adaptive educational systems. First, designer can implement complex event processing similar to the proposed adaptive language learning systems for the design of large-scale adaptive educational systems in the future. Educational big data including learner's static characteristics like age, learner's dynamic characteristics like learning context [50], learning mood [51], and learner feedback [52] can be captured and used to improve learning performance. Second, designer of adaptive educational system should shift to real-time learning behaviour patterns identification [53] such as tracking bad language habit forming and pay more attention to the dynamic learning behaviour analysis instead of the batch processing adaptation technique. Lastly, stream processing technologies provide adaptive educational system the opportunity to make fine-tuning in the adaptation process. With dynamic learning analysis [54] in pedagogy and behavior tracking using complex event processing, adaptive language learning can make adjustment such as rest warning, image presenting and other recommendation according to real-time learning behaviors.

Modeling learning interaction events and adjusting accordingly to language learning state is a complex task. Future investigations can explore pedagogy and linguistics complex event processing by extending the current system through rule managers and test the effectiveness of the new model. In the future, we will explore more pedagogical theories, simulation parameters, and interview language teaching professionals to extract more rules and increase the system's effectiveness. In additional, complex event processing over probabilistic flow is helpful in fuzzy learning scenarios. This will further benefit the adaptation process with fuzzy learning interaction events instead of deterministic events.

8. Acknowledgement

This work was supported in part by the National Social Science Foundation of China under Grant 13CTJ003 and in part by the China Postdoctoral Science Foundation under Grant 2014M562025.

References

References

[1] S. Sosnovsky, P. Brusilovsky, Evaluation of topic-based adaptation and student modeling in quizguide, User Modeling and User-Adapted Interaction 25 (4) (2015) 371–424.

[2] K.-U. Schmidt, R. Stühmer, J. Dörflinger, T. Rahmani, S. Thomas, L. Stojanovic, Adaptive reactive rich internet applications, in: Web 2.0 & Semantic Web, Springer, 2010, pp. 79–102.

[3] H.-C. Huang, T.-Y. Wang, F.-M. Hsieh, Constructing an adaptive mobile learning system for the support of personalized learning and device adaptation, Procedia-Social and Behavioral Sciences 64 (2012) 332–341.

[4] J. Murphy, J. B. Tracey, L. Horton-Tognazzini, Mooc camp: A flipped classroom and blended learning model, in: Information and Communication Technologies in Tourism 2016, Springer, 2016, pp. 653–665.

[5] R. Eynon, The rise of big data: what does it mean for education, technology, and media research? (2013).

[6] H. Lu, Y. Li, S. Mu, D. Wang, H. Kim, S. Serikawa, Motor anomaly detection for unmanned aerial vehicles using reinforcement learning, IEEE Internet of Things Journal.

[7] H. Lu, Y. Li, M. Chen, H. Kim, S. Serikawa, Brain intelligence: Go beyond artificial intelligence, Mobile Networks and Application (2017) 1–10.

[8] L. J. Fülöp, G. Tóth, R. Rácz, J. Pánczél, T. Gergely, Á. Beszédes, L. Farkas, Survey on complex event processing and predictive analytics, in: Proceedings of the Fifth Balkan Conference in Informatics, 2010, pp. 26–31.

[9] G. Cugola, A. Margara, Processing flows of information: From data stream to complex event processing, ACM Computing Surveys (CSUR) 44 (3) (2012) 15.

[10] A. J. Demers, J. Gehrke, B. Panda, M. Riedewald, V. Sharma, W. M. White, et al., Cayuga: A general purpose event monitoring system., in: CIDR, Vol. 7, 2007, pp. 412–422.

[11] J. Agrawal, Y. Diao, D. Gyllstrom, N. Immerman, Efficient pattern matching over event streams, in: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, ACM, 2008, pp. 147–160.

[12] H. Zhang, Y. Diao, N. Immerman, On complexity and optimization of expensive queries in complex event processing, in: Proceedings of the 2014 ACM SIGMOD international conference on Management of data, ACM, 2014, pp. 217–228.

[13] S. A. Douglass, –learner models in the large-scale cognitive modeling (lscm) initiative, Design Recommendations for Intelligent Tutoring Systems (2013) 111.

[14] W. Yao, C.-H. Chu, Z. Li, Leveraging complex event processing for smart hospitals using rfid, Journal of Network and Computer Applications 34 (3) (2011) 799–810.

[15] F. Terroso-Sáenz, M. Valdés-Vela, C. Sotomayor-Martínez, R. Toledo-Moreo, A. F. Gómez-Skarmeta, A cooperative approach to traffic congestion detection with complex event processing and vanet, IEEE Transactions on Intelligent Transportation Systems 13 (2) (2012) 914–929.

[16] Y. Xiao, W. Li, M. Siekkinen, P. Savolainen, A. Ylä-Jääski, P. Hui, Power management for wireless data transmission using complex event processing, IEEE Transactions on Computers 61 (12) (2012) 1765–1777.

[17] C. Bo, Z. Peng, Z. Da, C. Junliang, The complex alarming event detecting and disposal processing approach for coal mine safety using wireless sensor network, International Journal of Distributed Sensor Networks.

[18] S. Stipkovic, R. Bruns, J. Dunkel, Pervasive computing by mobile complex event processing, in: e-Business Engineering (ICEBE), 2013 IEEE 10th International Conference on, IEEE, 2013, pp. 318–323.

[19] C. Jun, C. Chi, Design of complex event-processing ids in internet of things, in: Measuring Technology and Mechatronics Automation (ICMTMA), 2014 Sixth International Conference on, IEEE, 2014, pp. 226–229.

[20] X. Qiao, B. Wu, Y. Liu, Z. Xue, J. Chen, Event-driven soa based district heating service system with complex event processing capability, International Journal of Web Services Research (IJWSR) 11 (1) (2014) 1–29.

[21] K. Jayan, A. K. Rajan, Preprocessor for complex event processing system in network security, in: Advances in Computing and Communications (ICACC), 2014 Fourth International Conference on, IEEE, 2014, pp. 187–189.

[22] J. Rimland, M. Ballora, Using complex event processing (cep) and vocal synthesis techniques to improve comprehension of sonified human-centric data, in: SPIE Sensing Technology+ Applications, International Society for Optics and Photonics, 2014, pp. 912203–912203.

[23] F. Terroso-Sáenz, M. Valdés-Vela, F. Campuzano, J. A. Botia, A. F. Skarmeta-Gómez, A complex event processing approach to perceive the vehicular context, Information Fusion 21 (2015) 187–209.

[24] J. Choi, R. Verma, S. Malhotra, Achieving fast operational intelligence in nasa's deep space network through complex event processing, in: 14th International Conference on Space Operations, 2016, p. 2375.

[25] S. Costantini, G. De Gasperis, G. Nazzicone, Dali for cognitive robotics: Principles and prototype implementation, in: International Symposium on Practical Aspects of Declarative Languages, Springer, 2017, pp. 152–162.

[26] D. Luckham, The power of events, Vol. 204, Addison-Wesley Reading, 2002.

[27] E. Alevizos, A. Skarlatidis, A. Artikis, G. Paliouras, Complex event recognition under uncertainty: A short survey, Event Processing, Forecasting and Decision-Making in the Big Data Era (EPForDM) (2015) 97–103.

[28] A. M. Pernas, J. P. M. de Oliveira, Enabling situation-aware behavior in web-based learning systems, in: Computer Science Society (SCCC), 2011 30th International Conference of the Chilean, IEEE, 2011, pp. 153–160.

[29] S. Cassidy*, Learning styles: An overview of theories, models, and measures, Educational psychology 24 (4) (2004) 419–444.

[30] Y. Akbulut, C. S. Cardak, Adaptive educational hypermedia accommodating learning styles: A content analysis of publications from 2000 to 2011, Computers & Education 58 (2) (2012) 835–842.

[31] Ö. Özyurt, H. Özyurt, Learning style based individualized adaptive e-learning environments: Content analysis of the articles published from 2005 to 2014, Computers in Human Behavior 52 (2015) 349–358.

[32] C. Hazotte, H. Mayer, Y. Djaghloul, T. Latour, P. Sonnleitner, M. Brunner, U. Keller, E. Francois, R. Martin, The genetics lab: An innovative tool for

assessment of intelligence by mean of complex problem solving, Informatics Engineering and Information Science (2011) 296–310.

[33] Z. L. He, Y. H. He, Key technologies of speech services in the internet era, in: Applied Mechanics and Materials, Vol. 427, Trans Tech Publ, 2013, pp. 2201–2205.

[34] S. M. Williams, D. Nix, P. Fairweather, Using speech recognition technology to enhance literacy instruction for emerging readers, in: Fourth International Conference of the Learning Sciences, 2000, pp. 115–120.

[35] L. Neumeyer, H. Franco, M. Weintraub, P. Price, Automatic text-independent pronunciation scoring of foreign language student speech, in: Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on, Vol. 3, IEEE, 1996, pp. 1457–1460.

[36] S. M. Witt, S. J. Young, Phone-level pronunciation scoring and assessment for interactive language learning, Speech communication 30 (2) (2000) 95–108.

[37] J. Burstein, K. Kukich, S. Wolff, C. Lu, M. Chodorow, L. Braden-Harder, M. D. Harris, Automated scoring using a hybrid feature identification technique, in: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1, Association for Computational Linguistics, 1998, pp. 206–210.

[38] Y. Attali, J. Burstein, Automated essay scoring with e-rater® v. 2, The Journal of Technology, Learning and Assessment 4 (3).

[39] K. W. Murray, N. Orii, Automatic essay scoring (2012).

[40] P. W. Foltz, D. Laham, T. K. Landauer, The intelligent essay assessor: Applications to educational technology, Interactive Multimedia Electronic Journal of Computer-Enhanced Learning 1 (2) (1999) 939–944.

[41] L. M. Rudner, T. Liang, Automated essay scoring using bayes' theorem, The Journal of Technology, Learning and Assessment 1 (2).

[42] H. Yannakoudakis, T. Briscoe, B. Medlock, A new dataset and method for automatically grading esol texts, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, 2011, pp. 180–189.

[43] P. Karampiperis, D. Sampson, Adaptive learning resources sequencing in educational hypermedia systems, Educational Technology & Society 8 (4) (2005) 128–147.

[44] M. R. Mendes, P. Bizarro, P. Marques, A performance study of event processing systems, in: Technology Conference on Performance Evaluation and Benchmarking, Springer, 2009, pp. 221–236.

[45] S. Kuboi, K. Baba, S. Takano, K. Murakami, An evaluation of a complex event processing engine, in: Advanced Applied Informatics (IIAIAAI), 2014 IIAI 3rd International Conference on, IEEE, 2014, pp. 190–193.

[46] C. Esposito, M. Ficco, F. Palmieri, A. Castiglione, A knowledge-based platform for big data analytics based on publish/subscribe services and stream processing, Knowledge-Based Systems 79 (2015) 3–17.

[47] Z. Dörnyei, Researching complex dynamic systems:retrodictive qualitative modellingin the language classroom, Language Teaching 47 (1) (2014) 80–91.

[48] C. Kulkarni, K. P. Wei, H. Le, D. Chia, K. Papadopoulos, J. Cheng, D. Koller, S. R. Klemmer, Peer and self assessment in massive online classes, in: Design thinking research, Springer, 2015, pp. 131–168.

[49] R. S. Baker, P. S. Inventado, Educational data mining and learning analytics, in: Learning analytics, Springer, 2014, pp. 61–75.

[50] S. Gómez, P. Zervas, D. G. Sampson, R. Fabregat, Context-aware adaptive and personalized mobile learning delivery supported by uolmp, Journal of King Saud University-Computer and Information Sciences 26 (1) (2014) 47–61.

[51] T. Gregersen, P. D. Macintyre, M. D. Meza, The motion of emotion: Idiodynamic case studies of learners' foreign language anxiety, The Modern Language Journal 98 (2) (2014) 574–588.

[52] F. M. Van der Kleij, R. C. Feskens, T. J. Eggen, Effects of feedback in a computer-based learning environment on students learning outcomes: A meta-analysis, Review of educational research 85 (4) (2015) 475–511.

[53] Y. Han*, S. Wei, S. Zhang, Analysis of online learning behaviour from a tutor perspective: Reflections on interactive teaching and learning in the big data era1, Asian Association of Open Universities Journal 10 (2) (2015) 29–48.

[54] S. Freitas, D. Gibson, C. Du Plessis, P. Halloran, E. Williams, M. Ambrose, I. Dunwell, S. Arnab, Foundations of dynamic learning analytics: Using university student data to increase retention, British Journal of Educational Technology 46 (6) (2015) 1175–1188.

Guolin Pan is currently working toward the Ph.D. degree at the School of Economics and Management, China University of Geosciences, Wuhan, China. He has participated in multiple scientific research projects at national and ministerial level. His research interests include big data, machine learning, learning systems, etc.

Si Shi received the M.Sc. degree major in information management and information systems, and minor in accountancy in 2014. She is currently pursuing the master's degree in management science and engineering with the Zhongnan University of Economics and Law, Wuhan, China. She has participated in multiple scientific research projects at national and ministerial level. Her research interests include real-time computing, complex event processing, data stream processing, financial time series, high-frequency trading, and algorithmic trading. Ms. Shi was a recipient of National Scholarship, the Outstanding Graduate Award, and First-class Academic Scholarship.

Dawei Jin received the bachelor's and master's degrees in management science from the Zhongnan University of Economics and Law, Wuhan, China, and the Ph.D. degree in computer science from Wuhan University, Wuhan. He visited the School of Computer Science, Deakin University, Australia, funded by the China Scholarship Council from 2011 to 2012. He is currently a Professor and a Ph.D. Supervisor with the Zhongnan University of Economics and Law. His current research interests include financial information engineering, financial high-frequency data analysis, high-frequency trading, and algorithmic trading.

Haider Abbas is Senior Member IEEE and Cyber Security Professional who took professional trainings and certifications from Massachusetts Institute of Technology (MIT), United States, Stockholm University, Sweden, IBM and EC-Council. He received his MS in Engineering and Management of Information Systems (2006) and PhD in Information Security (2010) from KTH-Royal Institute of Technology, Sweden. He is the Associate Editor of Electronic Commerce Research (Springer), IEEE Access, Cluster Computing (Springer), KSII Transactions on Internet and Information Systems, SpringerPlus, Ad Hoc \& Sensor Wireless Networks and Member Editorial Board for International Journal of Information Security and Privacy. He has also served as Lead Guest Editor for Peer to Peer Networking and Applications (Springer) and Annals of Telecommunications (Springer).

Yin Zhang is an Assistant Professor of the School of Information and Safety Engineering, Zhongnan University of Economics and Law (ZUEL), China. He is an IEEE Senior Member since 2016. He is an Excellent Young Scholar at ZUEL. He is Vice-chair of IEEE Computer Society Big Data STC. He was a Poster-Doctoral Fellow in the School of Computer Science and Technology at Huazhong University of Science and Technology, China. He serves as editor or associate editor for IEEE Access, IEEE Sensors Journal, etc. He is a Guest Editor for Mobile Networks and Applications, Sensors, Multimedia Tools and Applications, Journal of Medical Systems, New

Review of Hypermedia and Multimedia, etc. He also served as Track Chair of IEEE CSCN 2017, TPC Co-Chair of CloudComp 2015 and TRIDENTCOM 2017, etc. He has published more than 50 prestigious conference and journal papers. His research interests include intelligent service computing, big data and social network, etc.

Guolin Pan

Si Shi

Dawei Jin

Haider Abbas

Yin Zhang

- This paper presented a novel approach using complex event processing as a behavior tracker to process learning interaction events and provide real-time fine-tuning in an adaptive language learning system.
- The proposed approach is a pioneering work that CEP is adopted in language learning and adaptive educational systems.
- It is verified that the proposed approach is available to accelerate the learner's learning state tracking in language learning and improve the adaptive fine-tuning process.
- The performance evaluation shows that approach is feasible and effective.