Investigating multimodal affect sensing in an Affective Tutoring System using unobtrusive sensors

Hua Leong Fwa
Department of Computer Science
Newcastle University
h.l.fwa1@newcastle.ac.uk

Lindsay Marshall
Department of Computer Science
Newcastle University
Lindsay.marshall@newcastle.ac.uk

Abstract
Affect inextricably plays a critical role in the learning process. In this study, we investigate the multi-modal fusion of facial, keystrokes, mouse clicks, head posture and contextual features for the detection of student's frustration in an Affective Tutoring System. The results (AUC=0.64) demonstrated empirically that a multimodal approach offers higher accuracy and better robustness as compared to a unimodal approach. In addition, the inclusion of keystrokes and mouse clicks makes up for the detection gap where video based sensing modes (facial and head postures) are not available. The findings in this paper will dovetail to our end research objective of optimizing the learning of students by adapting empathetically or tailoring to their affective states.

1. Introduction
The critical role that emotion plays in learning is affirmed in a number of studies (Kort, Reilly, & Picard, 2001; Pekrun, Goetz, Titz, & Perry, 2002). Pekrun et al. (2002) conducted a series of quantitative and qualitative studies which concluded that positive affect e.g. enjoyment of learning affect achievement positively by strengthening motivation and enhancing flexible learning whereas negative affect such as anxiety and frustration erodes motivation and draws attention away from the task, resulting in shallow learning. These affirmed the role that affect plays in the learning process as well as the need to mitigate the effects of negative affect such as frustration.

Intelligent Tutoring Systems (ITSs) are built with the objective of providing learners with the benefit of one to one tutoring automatically and cost effectively by tailoring instructions to individual learning needs (Psotka, Massey, & Mutter, 1988). It acts as a personal training assistant that continually assesses one's knowledge through interactions with the system and builds a personalized model of one's acquired knowledge for the provision of tailored instructions or assistance in the form of hints or demonstrations when one seems to require help to move on. Some studies have in fact shown that ITSs outperform traditional classroom instruction in some domains (Anderson, Corbett, Koedinger, & Pelletier, 1995). However, for most domains, ITSs still underperform one-to-one tutoring. This has been attributed to the lack of emotional awareness in ITSs (Alexander, 2004; Picard, 1997).

Affective Tutoring Systems (ATSs) are Intelligent Tutoring Systems (ITSs) that adapt their tutoring responses based on the sensed emotions of the students, resulting in enhanced tutoring outcomes. The first step to building an ATS is to equip it with the ability to sense the emotions of the tutee. There are various affect sensing techniques and among the many, sentic modulation (Picard, 1997) is the technique which shows much promise in dynamic affect sensing. Sentic modulation refers to the physical assessment of a person's emotional changes via sensors such as cameras, microphones and wearable devices that register subtle physical modulation produced by emotional states.

Most previous studies on the use of sentic modulation have used a single sensor (unimodal) for affect sensing. Humans, on the other hand express our affect in multiple channels e.g. facial expressions, body postures and vocal intonations which thus leads to the belief that the use of multiple modalities for affect detection would more closely emulate human affect expression.Multimodal affect sensing is hypothesized to be superior to unimodal affect sensing as it is commonly believed that the multiple sensors complement one another. It also offers the affordance of data missing in some of the modalities as other modalities can make up for the missing data albeit at a degraded performance.

As compared to unimodal affect detection, the use of multiple modalities involves more technical complexities and issues. Baltrušaitis, Ahuja, and Morency (2018) aptly summarizes the technicalities

of multimodal affect detection into the 5 categories – Representation, Translation, Alignment, Fusion and Co-learning. Representation refers to the representation of heterogeneous multimodal data for the exploitation of complementarity and redundancies in multiple modalities. Translation refers to the mapping of data from one modality to another. Alignment refers to the identification of relations between elements from two or more modalities. Fusion refers to the joining of information from the modalities to perform a prediction while Co-learning relates to how knowledge learned on one modality can be transferred to a computational model trained on another modality.

Keyboards and mouse are standard input devices attached to every computer in a computer lab. Web cameras though not as common, are cheap and can easily be setup for use. Conceivably, these would qualify as economical commodity devices that are unobtrusive and thus suited for capturing the behavioural attributes of users inconspicuously.

In this study, we detail the techniques that we adopted to address some of the issues relating to multimodal affect sensing and to investigate into whether a multimodal and unobtrusive array of keystrokes, mouse clicks, facial features, head pose and contextual logs enhances the accuracy of affect sensing as compared to unimodal means.

2. Methodology

2.1. Participants

This work was compiled from data gathered in the trial conducted in Nanyang Polytechnic, Singapore in the year 2014 and 2015. The study was conducted in computer labs where 22 students were enrolled to work on programming exercises for an average period of 72 minutes in a Java programming tutoring software. Before the start of the session, students were requested to fill in a form granting consent to participate in the study. The students were also briefed on the objectives of the study and what was required of them in the trial. They were also guided on the various functions within the tutoring system.

2.2. Tutoring System

The students were provided with a total of 12 exercises to be completed within the tutoring software. Each exercise was preloaded with a set of Java codes with missing lines in between and students will have to fill in the missing lines of code to complete the exercise. Within each exercise page, students can click on a “Submit code” button to submit the code for compilation. They can then check the output window for the compilation output or errors. The compilation output will be verified against the required output when the student clicks on the “Check answer” button and will be marked as completed if the correct output is obtained.

The students’ actions, keystrokes and the respective time stamps in which each action or keystrokes occurred within the tutoring software were recorded. Some examples of the actions that were logged include the start and end time of each exercise, the time stamp of each “Submit code” action and the time when the exercise was completed. Copying and pasting of code from other web pages outside of the tutoring system was a common behaviour among novice programmers and that was captured in the keystroke logs as well.

2.3. Annotation of frustration

The face and screen video recordings were used for a retrospective annotation of frustration by two lecturer observers with an average teaching experience of five years. This retrospective observation technique was employed by a number of prior studies (Cetintas, Si, Xin, & Hord, 2010; Graesser et al., 2006; Mcquiggan, Lee, & Lester, 2007). Both the facial and screen video recordings were used for the annotation of frustration as it is difficult to ascertain whether the student is frustrated from facial expressions alone. The addition of screen video enhances the annotation accuracy by providing an additional source of information into the cause of the student’s frustration. For example, if the observer observed signs of frustration from the facial video but was not so sure, the observer could then confirm that the student was indeed frustrated (from the screen video) if for example, he or she observed that the student had been trying to rectify a compilation error for quite some time without success.

Some examples of the frustration behaviours noted in the session include use of expletives, long sighing, excessive gesturing and roughly ruffling through hair while visibly distressed. The observers find that these behaviours are usually accompanied with the encountering of compilation errors or being stuck in a particular point in the exercise for a period of time (which can be observed from the students' screen videos). The observations were recorded with a time stamp which was used for synchronizing with the captured contextual and keystroke logs.

The average length of each student's video segment is 72.3 minutes (with a standard deviation of 10 minutes) making up a total of n=9502 instances (overlapping time window slices using the sliding window mechanism) in the data set. There was an average of 15 instances of frustration (with a standard deviation of 6) noted per student's video segment.

2.4. Features

The facial features are captured from web cameras installed on top of the desktop monitors. The iMotions FACET software (iMotions, 2017), a commercial version of the Computer Expression Recognition Tool (CERT) was used for extracting likelihood estimates of 17 Action Units (AUs) from the captured videos. These 17 AUs denote facial muscles movements of the brow, eye lid, nose and lip. From these 17 AUs' likelihood estimates, we derived additional features by calculating the median, maximum and standard deviation of each, making a total of 51 features. These 51 features are temporally aligned with frustration observation in the 30 seconds time window.

The list of contextual features includes the number of exercises completed, the number of submissions for compilation of the exercise, the number of switches between the exercises, exercise worked on, exercise duration and the number of errors encountered. The number of exercises completed denotes the number of exercises that the student has completed. The number of submissions for compilation denotes the number of times the student has submitted the code for compilation for the designated exercise. During the course of the study, we observed that students click on the submit for compilation button several times within a span of two seconds, thinking that more clicks will help to speed up the compilation time. Thus, to prevent duplicate counting, all submissions for compilation logs (by the same student for the same exercise) time-stamped within duration of two seconds are only counted as one submission for compilation.

The duration or flight time of the key is the duration from the time a key is depressed to the time when the next key is depressed. The keystroke log files on the server for all the students are consolidated and processed using a batch program to calculate the mean, median and frequency of the different groups of keys (e.g. alphanumeric keys, navigation keys, backspace keys e.t.c.). The duration between keys refers to the flight time (the duration from the time one key is depressed to the time the next key is depressed) and the wait duration refers to the duration in which no key was depressed.

Mouse clicks were captured only when the students were working on the exercises. A JavaScript function running on the client end captures and sends the raw mouse clicks information to the server. At the server end, a program processes the raw mouse clicks information which consists of the coordinates of the mouse click, characteristics of the mouse click (single click or double clicks) and time stamp denoting the time of depression of the mouse click. This processed information is then written to a log file on the server. The logged mouse clicks files on the server are then consolidated and processed using a batch program to derive the total number of clicks, number of clicks less than 2 seconds and the number of double clicks.

The head posture features were captured from web cameras installed on the top of the desktop monitors. An eye tracking software from xLabs (xlabsgaze.com) was used for extracting raw head pose information such as horizontal and vertical head position, head roll, pitch and yaw from the captured videos. These were further processed to derive the median, standard deviation and maximum position and the pitch, roll and yaw velocities. The pitch, roll and yaw velocities were derived from the differences of the current pitch, roll and yaw values from the pitch, roll and yaw values of the previous second.

The facial, head pose, contextual, keystrokes and mouse clicks features were lastly combined into a single channel feature vector that is passed to the classifiers for classification.

2.4. Synchronization of features with annotations

The students' facial, head postures, contextual, keystrokes and mouse logs were aggregated into features using a sliding window size of 30 seconds with an overlap of two-thirds of the window size. If the time at which frustration was observed falls in the overlap area of 2 consecutive window slices, both window slices would be annotated as slices in which the student experiences frustration. Alternatively, if the time at which frustration was observed falls outside the overlap area, only the time window slice in which it occurred in will be annotated as the slice in which the student experienced frustration. This is illustrated in Figure 1.

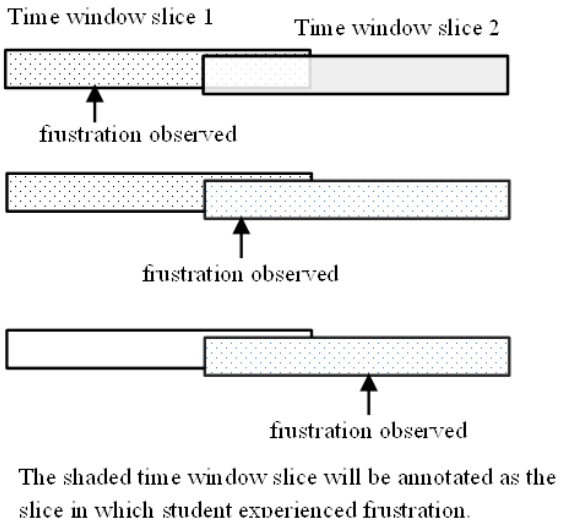

Figure 1- Overlapping time window slices to synchronize features with annotations

2.5. Features Selection and Classification

The frustration detection classification models in this study were built using the 3 separate channels (facial channel, contextual, keystrokes and mouse clicks channel and head pose channel), feature fusion (2 channels and 3 channels) and decision fusion (using max and mean vote), making up a total of 7 models.

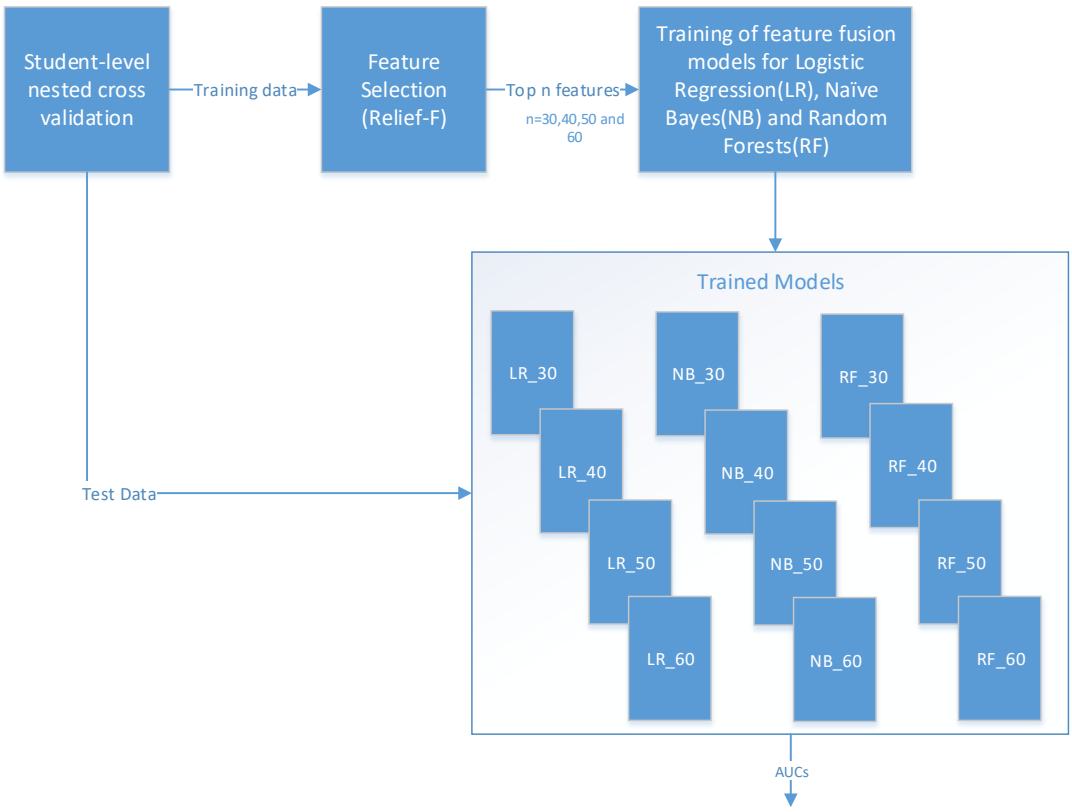

Figure 2 – Cross validation, feature selection and classification for feature fusion models

The 3 base channels are the channels that combine the contextual, keystrokes and mouse clicks, the facial channel and lastly the head pose channel. The feature fusion model combines the features from the 3 base channels into a combined feature vector for feeding into the classifiers. The decision fusion model uses the combination of the decision probabilities from the 3 base classifiers (built using the features of the 3 base channels' features respectively) for a decision on the final classification output. The final classification output for the decision fusion model can then be determined for instance, by taking the highest of the decision probabilities among the 3 classifiers.

The 3 classifiers used in the models were Random Forests (RF) (Breiman, 2001), Logistic Regression (LR) and Naïve Bayes (NB). The features are extracted from the facial, head posture, contextual, keystrokes and mouse logs. In total, 51 features are extracted from the facial channel, 40 features from the combined keystrokes, mouse clicks and contextual channel and 11 features from the head postures channel. For the feature fused model, 30, 40, 50 and 60 features selected using the RELIEF-F (Kononenko, 1994) feature selection algorithm are fed into the RF, LR and NB classifiers.

RELIEF-F algorithm first selects a random instance and determines the instances that are closest to the selected random instance using the Manhattan distance. The weights to features are then updated through reducing them by the features' absolute difference for instances that are close to each other and of the same class type and increasing them by the features' Euclidean distance for instances that are close to each other and of a different class type. The features are then ranked by weights and the top ranked n features are then selected for use in the classification.

The cross validation, feature selection and classification workflow for feature fusion models is shown in Figure 2.

2.5. Student-level nested cross validation

To ensure the generalizability of the classification models, all the models were tested using student-level nested cross validation. In the outer loop of the cross validation, the data set for a randomly selected 66.7% of the students were used as the training set with the data set for the remaining 33.3% of the students as the test set. In the inner loop, a further 66.7% of the student's data within the outer

training set (44.5% of the whole data set) were then used for feature selection. The feature selection results were then averaged over 10 runs of the inner loop. In addition, lasso regularization was applied for logistic regression with the lambda regularization value determined through 5-fold cross validation. The classification results for each of the models were averaged over 30 runs of the outer loop to derive the final classification results.

The Area under the Receiver Operator Characteristic Curve (AUC) is used as a performance measure to compare between the various models as it is useful for domains with skewed class distribution and unequal classification error costs (Fawcett, 2006). It is preferred here as compared to accuracy as a performance measure because instances of frustration are rare as compared to non-frustration. A naive classification model that always predict non-frustration for all test instances will have a high accuracy measure but low AUC measure as it does not discriminate instances of frustration from non-frustration.

3. Results

In this section, the classification results of the models for discriminating between instances of frustration from non-frustration for the different channels are reported. The AUCs for the facial channel (FC), head pose channel (HPC) and keystrokes, mouse clicks and contextual channel (KMC) using the Random Forest, Logistic Regression and Naive Bayes classifiers are shown in Table 1.

Channels	**Classifiers**			**No. of features**
	Random Forest	**Logistic Regression**	**Naive Bayes**	
Facial (FC)	0.552	**0.58**	0.555	25
Head Pose (HPC)	0.51	0.542	**0.553**	5
Keystrokes, Mouse clicks and Contextual (KMC)	0.539	**0.575**	0.5	20

Table 1- Classification results for various channels by classifiers

From the results, the facial channel offers the best performance (AUC=0.58) followed by the keystrokes, mouse clicks and contextual logs channel (AUC= 0.575). 25 facial features, 5 head pose features and 20 combined keystrokes, mouse clicks and contextual features are extracted for FC, HPC and KMC respectively using the RELIEF-F feature selection algorithm. The classifiers for each of the 3 channels are better than the random model with AUC=0.5, thus providing evidence that each of the 3 classifiers can discriminate between instances of frustration better than chance. It can also be seen from Table 1 that the logistic regression classifier offers the best performance among the 3 classifiers for FC and KMC.

The classification results for the various fusion models that combine the channels are shown in Table 2. The 3 channels feature fusion model combines the features for the FC, HPC and KMC into a large feature vector for classification. The 2 channels feature fusion model combines the features for the FC and HPC into a large feature vector for classification. A range of features from 30, 40, 50 and 60 features are extracted using the RELIEF-F algorithm for the feature fusion models. For the decision fusion (max) model, the maximum of the decision probabilities from the base classifiers is used as the final decision output of the model. For the decision fusion (sum) model, the decision probabilities from the base classifiers are averaged to determine the final decision output of the model.

Fusion Models	**No. of features**	**AUC (Logistic Regression)**
3 channels feature fusion	30	**0.636**
2 channels feature fusion	30	0.631
3 channels feature fusion	40	0.621
2 channels feature fusion	40	0.607
3 channels feature fusion	50	0.582

2 channels feature fusion	50	0.58
3 channels feature fusion	60	0.568
2 channels feature fusion	60	0.568
Decision Fusion (Max)	FC:25, HPC:5, KMC:20	0.578
Decision Fusion (Average)	FC:25, HPC:5, KMC:20	0.588

Table 2 – Classification results for the various fusion models using Logistic Regression

The results show that the 3 channels feature fusion model using 30 features has the best performance (AUC=0.636) among the fusion models. The AUC of 0.636 of the feature fusion model is 9.7% higher than the best unimodal channel (facial channel) with an AUC of 0.58, verifying that multimodal fusion leads to higher detection accuracy over unimodal model. In general, the feature fusion models perform better than the decision fusion models (for those feature fused models with 50 or lesser features).

The AUC of the 2 channels feature fusion model which combines FC and HPC is only slightly lower than that of the 3 channels feature fusion model. This shows that the inclusion of keystrokes, mouse clicks and contextual channel in the 3 channel model only slightly enhanced the classification performance compared to that of the 2 channel model that includes only facial and head pose channel features. However, it is still relevant to include keystroke, mouse clicks and contextual features in the fusion model as the facial and head pose features are unavailable for an average of 17% of the total sessions across all students. If we deploy this tutoring system in an actual classroom environment where lighting and occlusion issues are more prevalent, the availability of the facial and head pose channels will be further reduced. Thus, the addition of keystrokes, mouse clicks and contextual channel features complements the affect detection using facial and head pose channel features.

3. Conclusion

The main goal of this study is to explore automated techniques for the detection of frustration in a naturalistic learning environment. With adequate detection of frustration on a moment by moment basis, hints and tutorial supports can be provided to the students to overcome learning barriers and alleviate their frustration so as to sustain their engagement in learning.

In this study, we have explored the use of unobtrusive sensors in affect detection for the context of a tutoring system that tutors students in programming. More specifically, we have established the viability of using keystrokes, mouse clicks and contextual logs for the detection of frustration on a level of granularity that is adequate for timely remedial intervention.

Keystrokes and mouse clicks are traditionally used in computer security domain for authentication and user identification purposes and are rarely used for affect detection and thus, the significance of the use of keystrokes and mouse clicks for affect detection in this study. Importantly, we are also confident of the generalizability of our results owing to the use of student-level nested-cross validation for validating the models.

Another area of significance is in multimodal affect detection – the fusing of multiple sensing modes outputs. It is a fact that human emotion is expressed in various channels e.g. facial, vocal and bodily expressions but implementation of a multimodal affect detection system is still rare in occurrence (Jaimes & Sebe, 2007). In this study, a multimodal system of affect detection using keystrokes, mouse clicks, contextual logs, facial and head postures combined using both feature fusion and decision fusion techniques is proposed and implemented. It is further verified that a multimodal fusion of the proposed sensors does outperform the best unimodal channel (the facial channel). Although the features that contribute most to the accuracy of the multimodal model are the ones that are derived from head postures and facial channels, the keystrokes and mouse clicks do make up for the periods of detection gaps where both the head postures and facial features are not available.

An extension of this study will be to implement the above described affect detection technique in a programming tutoring system. Augmenting the tutoring system with the ability to infer the affect of

students is the first step in the construction of an ATS. The challenge consequent to this will be to design the ATS such that it can respond appropriately to the detected emotions of the students to increase engagement, task persistence and sustain their motivation to learn. This will entail the tailoring of various affect-driven and pedagogical focused measures e.g. hints and empathetic supports to enhance the tutoring effectiveness of the ATS, thus effectively closing the loop of affect detection and intervention.

4. References

Alexander, S. (2004). *Emulating human tutor empathy.* Paper presented at the Proceedings of the IIMS Postgraduate Conference, Albany, New Zealand.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences, 4*(2), 167-207.

Baltrušaitis, T., Ahuja, C., & Morency, L.-P. (2018). Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Breiman, L. (2001). Random forests. *Machine learning, 45*(1), 5-32.

Cetintas, S., Si, L., Xin, Y. P., & Hord, C. (2010). Automatic detection of off-task behaviors in intelligent tutoring systems with machine learning techniques. *IEEE Transactions on Learning Technologies, 3*(3), 228-236. doi: 10.1109/tlt.2009.44

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters, 27*(8), 861-874. doi: 10.1016/j.patrec.2005.10.010

Graesser, A., McDaniel, B., Chipman, P., Witherspoon, A., D'Mello, S., & Gholson, B. (2006). *Detection of emotions during learning with AutoTutor.* Paper presented at the Proceedings of the 28th Annual Meetings of the Cognitive Science Society.

iMotions. (2017). API. from https://imotions.com/api/

Jaimes, A., & Sebe, N. (2007). Multimodal human-computer interaction: A survey. *Computer Vision and Image Understanding, 108*(1-2), 116-134.

Kononenko, I. (1994). *Estimating attributes: analysis and extensions of RELIEF.* Paper presented at the European conference on machine learning.

Kort, B., Reilly, R., & Picard, R. W. (2001). *An affective model of interplay between emotions and learning: Reengineering educational pedagogy-building a learning companion.* Paper presented at the Advanced Learning Technologies, IEEE International Conference on.

Mcquiggan, S. W., Lee, S., & Lester, J. C. (2007). Early prediction of student frustration *Affective Computing and Intelligent Interaction* (pp. 698-709): Springer.

Pekrun, R., Goetz, T., Titz, W., & Perry, R. P. (2002). Academic emotions in students' self-regulated learning and achievement: A program of qualitative and quantitative research. *Educational psychologist, 37*(2), 91-105.

Picard, R. W. (1997). *Affective computing* (Vol. 252): MIT press Cambridge.

Psotka, J., Massey, L. D., & Mutter, S. A. (1988). *Intelligent tutoring systems: Lessons learned*: Psychology Press.