Can Your Plane Tweet? An Experiment in User Performance Tracking

Brandt Dargue, Cliff Sowadski
Boeing Research & Technologies
St. Louis, MO
Brandt.W.Dargue@Boeing.com, Clifford.B.Sowadski@Boeing.com

ABSTRACT

Collecting and storing user actions and decisions during training and job performance is important for accurate records and learner assessments. The specifications and standards that exist for formatting, sending, storing, and accessing that information are designed to be used with a web-based Learning Management System (LMS) that launches web-based training content. Therefore, they are not suitable for simulation-based training, live training, learning in serious games, on the job training (OJT), and informal learning.

The Advanced Distributed Learning (ADL) initiative has proposed an “Experience Application Programmer’s Interface (API)” that sends experience tracking information to a Learning Record System (LRS) using the same mechanisms that are used to send brief social messages using web pages or mobile phones. Boeing Research & Technologies conducted research to determine if the Experience API could be used with high-fidelity flight simulations and live flight. For this study, we created a prototype of an on-board adaptive live fighter training scenario which used of the Experience API (xAPI) to send student performance data to an open source LRS. We also built a custom report for the LRS showing a “leader board” of the top ten scores. The prototype was featured on the exhibit floor at I/ITSEC 2012 and the innovative use of the Experience API and LRS, which turned training into a fun game, was very well received.

This paper will discuss the prototype automated assessment technology and the “Tweets” to the LRS using the Experience API. We will discuss lessons learned and provide suggestions for the future. We will discuss how the same technology and API can be used to track much more than training performance records. For example, we will discuss how it can be used for flight ramp maintenance audit trails or even to automatically provide credit for conducting tutorials or presentations like this.

ABOUT THE AUTHORS

Brandt Dargue is an Associate Technical Fellow performing research into current and future training technologies including simulations, automated performance assessment, adaptive scenarios, intelligent tutoring, virtual environments, mobile platforms, gaming concepts and gaming technologies. He has been employed at Boeing for 24 years, designing, developing, integrating, and testing systems and content for aircrew and maintenance training. Brandt has chaired or participated in several international standards development and study groups. Brandt invented the Boeing Intelligent Tutoring System (ITS) and co-invented the patented Embedded Instructor automated performance assessment and scenario adaptation tool, and has several other training-related patents pending. Brandt is the Program Manager and Principal Investigator for “The Enemy of Reason” contract with the Intelligence Advanced Research Projects Activity (IARPA) (contract FA8650-12-C-7234) to develop a game to teach Intelligence Analysts to recognize and avoid biased interpretation of information and the design/performance of associated game based training effectiveness studies.

Cliff Sowadski is a Software Engineer in Boeing Research & Technology (BR&T). He has 13 years of software engineering experience both inside and outside the Boeing Company. Cliff has worked in many areas throughout his career, from networking systems to simulations to gaming technologies. He co-invented and developed the patented Embedded Instructor automated performance assessment and scenario adaptation tool. His work includes an innovative authoring tool that allows subject matter experts to create complex adaptive training scenarios. He has been a judge in the Serious Games Challenge and Showcase for several years and his work has often been featured at the Boeing booth. Cliff is currently BR&T’s Principal Investigator for Advanced Adaptive Learning Research and Development.

Can Your Plane Tweet? An Experiment in User Performance Tracking

Brandt Dargue, Cliff Sowadski
Boeing Research & Technologies
St. Louis, MO
Brandt.W.Dargue@Boeing.com, Clifford.B.Sowadski@Boeing.com

INTRODUCTION

An Introduction

There are several items that should be introduced to the reader of this paper before getting into the details of the problem, approach, and results. First, we will introduce the overall goal of our effort – to make training more effective; and the immediate objective – to create records of training flights. Second, we will introduce how training records are currently created from training activities – by hand or, when automated, using interfaces defined by standards and specifications that Learning Management Systems LMSs) use. Third, we introduce why we needed to try a new approach. Then the primary body of the paper will discuss what the new approach was, whether it worked, and finally, how that approach can be used in the future.

The Goals

The primary goal to this project is quite simply to make training more effective. This includes making the learning activities more productive in order to make learners more productive so that they are more productive on the job. When the learners are fighter pilots, being more productive means they are creative while quick, decisive while flexible, and deadly while safe. Like most tasks that require a blend of intelligence and skills, practice is critical to gaining expertise in tactical air tasks (Ericsson, 2009). Our goal is to ensure that the practice is done in the best way possible. Blending the time-tested approaches of mentoring, apprenticeships, tutoring, coaching, and on-the-job training with state of the art simulations, adaptive learning/training, and the theory of flow (Csíkszentmihályi, 2008) significantly increases effectiveness. This new systems approach to learning creates a closed-loop, self-improving learning environment that provides the right amount of challenge in the most effective way to accelerate learning and makes it more impactful for the individual learner.

One of the obvious keys to adaptive systems is that they need to be adaptable – in other words adjusted as needed. The key enabler is the ability to monitor [Observe, detect, sense] and assess the performance of the system to determine if/when/what parameters are significant to assess and what corrections (adjustments, adaptations) should be used to optimize learning. This cycle is shown in the simplified Figure 1.

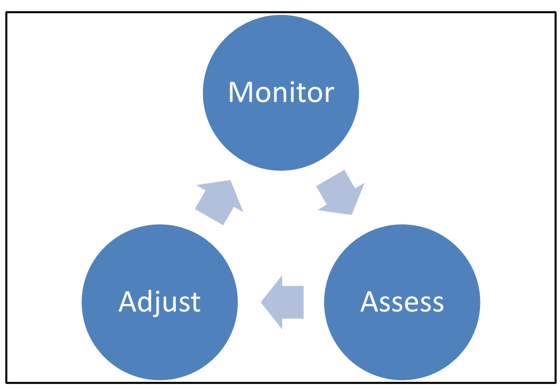

Figure 1 - Closed-Loop Adaptive System Cycle

Two important items are not included in the simplification of the figure. The individual learner and their training records are key in an individualized adaptive training system. Both the monitor and the assessor objects in the figure should be able to create individualized records of the training experience. These records are used for the closed-loop analysis, but also for analysis that compares different learners, instructors, approaches. There are standards, specifications, and reference models for the creation of training records that we have used in many projects. However, we required custom solutions when we applied individualized adaptive learning techniques and technologies to an integrated live, virtual, and constructive tactical air training prototype. These custom solutions posed a risk of not being reliable nor robust. Therefore, when the specifications, standards and reference model developers proposed a new specification to solve the problems, we added a test implementation of the specification to the prototype.

Creating Training Records – AICC, the SCORM, and More

Interoperability guidelines, recommendations, specifications, and standards for computer-based training (CBT) emerged shortly after the technology became practice. The Aviation Industry CBT Committee (AICC) was formed in 1988 to enable standardization of CBT hardware platforms (AICC Frequently Asked Questions, 2013). In 1993, AICC released a guideline (AGR-006) for Computer-Managed Instruction (CMI) interoperability to enable Learning Management Systems (LMSs) to manage and collect data from CBT courseware from multiple vendors. AICC guidelines and specifications for CMI were updated on a regular basis to support new technologies including the Internet, the Web, and web browsers.

The Advanced Distributed Learning (ADL) Initiative was started to meet a 1999 Presidential Executive Order to ensure federal agencies take full advantage of technological advances in order to acquire the skills and learning needed to succeed in an ever-changing workplace. One of the responsibilities of ADL is to assist in or direct the development of learning interoperability specifications. ADL is most commonly associated with the Shareable Content Object Reference Model (SCORM) which is a set of guidelines, specifications, and standards such as the CMI Data Model and the ECMA Script API to develop interoperable training content.

Of particular interest to recording and accessing learning records are the CMI Data Model and JavaScript API interface specifications. Both have become IEEE standards. The CMI or the "Data Model for Content to Learning Management System Communication (IEEE Standard for Learning Technology, 2005)" describes what training records look like. It defines the format, structure, rules, and contents including allowable values. The JavaScript API or "ECMA Script Application Programming Interface for Content to Runtime Services Communication" (IEEE Standard for Learning Technology, 2004) describes how the training records are communicated from the training content to the runtime system (e.g. LMS). It defines the API statements to "set" and "get" values for the CMI data using name/value pairs, statements to start and stop the communication, and statements for checking for and getting details on errors. As the name implies, the typical LMS manages the delivery of learning content as well as the learning records. In that role, the LMS provides for a learner to log into the secure system, makes learning records and plans available, allows the learner to register or be assigned to specific courses and/or lessons, launches the content for the learner, sequences the lessons as required, and collects/stores the CMI data. The SCORM specifies that when the LMS launches a shareable content object (SCO) in the web browser that it provides the API in such a way that JavaScript in the SCO can find it and use it. The particular states that the API and the SCOs must support are shown in Figure 2. Note that the API will not allow access to the CMI data until the SCO is launched and the SCO successfully invokes the Initialize("") method. These states and functionality work well for web-based content that is launched by the LMS and can use the JavaScript-based API.

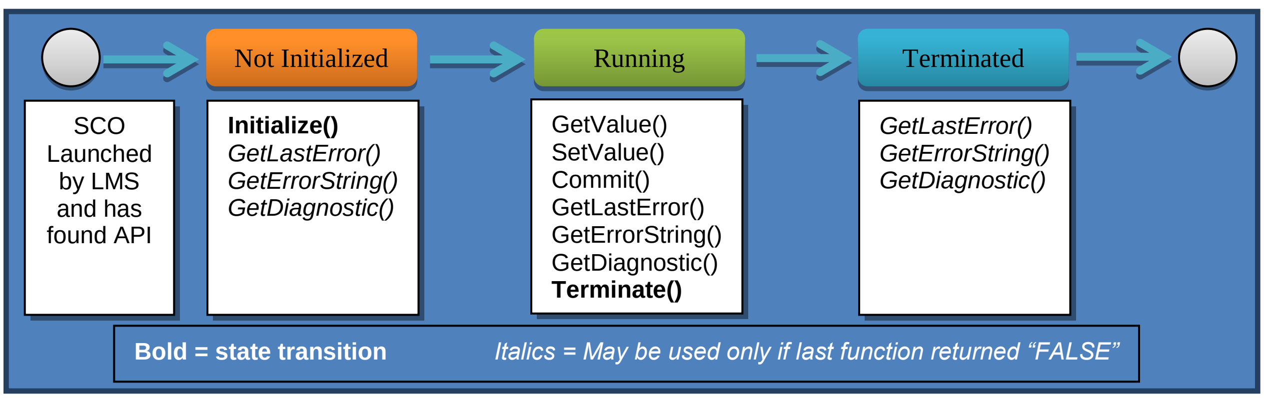

Figure 2 - API States, Functions, and Transitions

The Need for a New Approach

To perfect the required techniques in a highly skillful, fast-paced domain like tactical air, as the classic R&B song by Ashford & Simpson goes, there "Ain't Nothing Like the Real Thing." However, tactical air practice flights in "the real thing" require carefully choreographed, coordinated exercises involving many jets. One example of how amazingly well this works was the US Navy Fighter Weapons School (well known as Top Gun). This training method increased pilot effectiveness by a magnitude using "an instrumented range, better than real enemy, and no-holds-barred after-action reviews (AARs)" (Chatham, 2009). Boeing's integrated Live, Virtual, and Constructive (iLVC) technology leverages this approach while providing more efficient use of student flight time, increasing safety, increasing realism, and reducing cost. With iLVC, some of the real jets are replaced with constructive and virtual simulations, eliminating/reducing the maintenance and fuel costs of real jets. The onboard sensors of iLVC-enabled aircraft display these simulated virtual and constructive entities as if they were real. In this way, the aircrew are immersed in very realistic scenarios that would be dangerous, costly, and very difficult, (if not impossible) to perform using other live aircraft. The Boeing implementation approach for iLVC utilizes ground stations or other aircraft in "tethered" mode, or can run stand-alone, completely self-contained on the student's aircraft. In either mode, it is important to accurately evaluate the pilot and report the assessment back to the LMS and the AAR system(s).

While iLVC provides a more accurately reproduced scenario, it is still difficult to objectively access all the nuanced actions performed in such a fast-paced environment without automated performance assessment (Schreiber, 2009). Boeing uses an assessment engine that evaluates real-time automated performance assessment to optimize the learning experience of training exercises.

The standards of the SCORM are designed to enable training content that is launched by a web-based LMS and runs in a web browser to communicate with the LMS. To enable a training aircraft to be the training "content," custom solutions were required to communicate with the LMS. A simple solution implemented for flight simulators was to format the CMI data using XML as a file that is read by the LMS-launched SCO. The file could be encrypted and contain information to ensure integrity. The SCORM-compliant sequencing rules ensure that the SCO successfully loads the file and uploads the information to the LMS before any other training is permitted to be launched. This solution worked well in prototypes. Unfortunately, ensuring that the CMI data file remains accessible to the SCO proved problematic. First, the location must be available to both the simulator Instructor/Operator Station (IOS) and the browser-based SCO. It also involved manual steps where the student must manually log into the LMS and launch the course that has the SCO. When the SCO had to read in the file, if the file was not on the same server, there would be manual security override prompting.

Our proposed ideal solution allows any training software an any style of device to send training records to the LMS. Example software and devices include simulations, simulators, training aircraft, IOSs, debrief stations, or electronic grade sheets on mobile devices.

Other requirements for an ideal solution include the ability to:
[http://www.adlnet.gov/adl-summarizes-next-generation-scorm-requirements]

- Track diverse user learning experiences such as social media, mobile learning, games, simulations, and mixed modality exercises.
- Move beyond the single-learner model to allow team-based exercises, collaboration, and direct instructor intervention.
- Eliminate out-of-date technology practices such as the idea that all content needs to fit into a "package".
- Improve or eliminate the content sequencing and ineffective requirements.
- Include tools, guides, and best practices to make the learning curve for Next Generation SCORM easier.
- Provide clearer instructions and more efficient testing to make content work across systems.
- Allow content to function in situations where access to network infrastructure is limited or intermittent.
- Provide a means to expose user data to instructors while keeping assessment data secure.

THE EXPERIENCE API (AKA TIN CAN)

The ADL Initiative solicited requirements and recommendations for the next version of SCORM which resulted in the list of requirements above as well as numerous suggested approaches. Rustici Software was awarded a contract to analyze the papers and develop the solution. The project, codenamed Tin Can, recommended that to enable tracking activities that are relevant to training records, the activity records should be stored "in the cloud" using a

Learning Record Store (LRS) that can be accessed by software on any connected device or server using a simple API. The LRS provides the specified API to store and access the records as well as provide security and data integrity checks. The API is officially called the Experience API (xAPI), but the Tin Can API moniker is still used by many [http://tincanapi.com/2012/12/11/we-call-it-tin-can/].

The xAPI specification version 1.0 was released 29 April 2013 [https://github.com/adlnet/xAPI-Spec/blob/master/xAPI.md]. Draft specifications and open source sample software were used for the 2012 prototype discussed in this paper.

The Tin Can Project set out to find a very easy to use, easy to implement, secure method that runs on practically any device. They used Twitter as a model. Recent evidence of Twitter meeting a real world goal is the "TweetPee", a small clip that attaches to a diaper and sends a Twitter message to the baby's parent when their baby's diaper needs changing. The included app tracks diaper usage trends and home inventory to alert when more diapers need to be purchased. If a diaper clip can do it, why can't advanced aircraft and aircraft training systems?

The Tweet

While the xAPI is not identical to Twitter, it is based on simple messaging protocols used by SMS Text, Twitter, Facebook, Google, and Microsoft. To send an activity record to the LRS, an application sends secure statements in the form of "Actor, verb, object" or "I did this" using the xAPI. These statements are then put into the JavaScript Object Notation (JSON) format before sending to the LRS for storage. First introduced as a scripting language for the Netscape browser, JSON is now part of the ECMAScript standard as a way to serialize objects for sending as messages and is thus supported by all modern browsers.

In order to describe what an xAPI statement is, it would be best to start with an example and then break it down (Table 1). For this paper, we will use the statement: **Barb Completed Intercept Lesson 1**.

Table 1 - The elements of the xAPI format for the example message

Noun ("Barb") – This component uniquely identifies the actor for the statement.	"actor": {
	"name": "**Barb**",
	"mbox": "Barb@lrs.com"
	}

Verb ("Completed") – The verb component indicates what action occurred for this statement. The xAPI specifies that it be formatted as a URI. For example, if the "Actor" "completed" some a learning objective, the verb would be as follows:	"verb": {
	"id": "http://adlnet.gov/expapi/verbs/completed",
	"display": {"en-US" : "**completed**"}
	}

Object ("Intercept Lesson 1") – The object component identifies the specific activity in which the verb component occurred. For our example, if the student completed Lesson 1 for the Intercept course, the object component of the statement could be as follows.	"object": {
	"id": "Boeing.com/TCAPI_Intercept101-1",
	"definition": {
	"name": "Boeing Intercept 101 Lesson 1",
	"type": "Training",
	"description": "**Intercept Lesson 1**"
	}
	}

xAPI Vocabularies

In order to provide data normalization for interoperability and future analysis of the data, it is recommended that the verbs and object types are standardized in a common way. The xAPI handles this through profiles which define the vocabularies used for verbs and object types for each particular domain. For the learning/education/training domain, the profile development is being led by the AICC with support from ADL and other organizations. Repositories of the notional verb and activity type repository is http://adlnet.gov/expapi/ (also see https://registry.tincanapi.com/). For example, Table 2 shows the vocabulary defined by the ADL community for Activities. Table 3 shows the vocabulary defined by the ADL community and AICC. Note that there are some differences in ADL and AICC's lists in the tables. ADL's list is all inclusive of any learning activity whereas AICC is currently defining the interfaces between training content and LMSs. Note that the AICC approach for CMI-5 is that the LMS and the content will use the xAPI and the LRS as the communication mechanism. Therefore, the LMS will be required to enter data into the LRS that is needed by the content into the LRS for the content to retrieve.

Table 2 - Experience API Activity Vocabulary

Activity Type	URI	Description
Assessment	http://adlnet.gov/expapi/activities/assessment	Activity that determines a learner's mastery of a subject area. Typically has one or more questions.
Course	http://adlnet.gov/expapi/activities/course	Typically the largest level of granularity assigned.
Interaction	http://adlnet.gov/expapi/activities/interaction	An individual learner action or input.
Link	http://adlnet.gov/expapi/activities/link	Simply a means of expressing a link to another resource.
Media	http://adlnet.gov/expapi/activities/media	Passive text, audio, or video used to convey information.
Meeting	http://adlnet.gov/expapi/activities/meeting	A meeting is a gathering of multiple people for a common cause
Module	http://adlnet.gov/expapi/activities/module	Any "content aggregation" at least one level below the course level and one level up from all content.
Objective	http://adlnet.gov/expapi/activities/objective	A competency goal in a desired area.
Performance	http://adlnet.gov/expapi/activities/performance	An attempted task or series of tasks within a particular context.
Question	http://adlnet.gov/expapi/activities/question	Typically part of an assessment and requires a response from the learner, a response that is then evaluated for correctness.
Simulation	http://adlnet.gov/expapi/activities/simulation	Task or series of tasks in an artificial context that mimics reality. Tasks often take on the form of interactions.

Table 3 - Experience API Verb Vocabulary

Verb	ID	Description
abandoned	http://www.aicc.org/cmi5/verbs/abandoned	Indicates that the session was abnormally exited by Learner action (or due to a system failure).
completed	http://adlnet.gov/expapi/verbs/completed	The activity was experienced in its entirety. Any content that has been initialized, but not yet completed, should be considered incomplete. There is no verb to 'incomplete' an activity, one would void the statement which completes the activity.
exited	http://www.aicc.org/cmi5/verbs/exited	User left an activity attempt. Note that the draft ADL repository suggests that "exited" indicates that the user has no intention of returning, thus progress is discarded.
failed	http://adlnet.gov/expapi/verbs/failed	Learner did not perform the activity to a level of pre-determined satisfaction.
launched	http://www.aicc.org/cmi5/verbs/launched	Starts the process of launching the next piece of learning content. If the content was launched by an LMS, the LMS is responsible for sending this verb.
passed	http://adlnet.gov/expapi/verbs/passed	Affirms the success a learner experienced within the learning content in relation to a threshold.
resumed	http://adlnet.gov/expapi/verbs/resumed	Used to resume suspended attempts on an activity.
started	http://www.aicc.org/cmi5/verbs/started	Indicates that the content was started (e.g. launched by the LMS)
suspended	http://adlnet.gov/expapi/verbs/suspended	Used to suspend an activity with the intention of returning to it later, and not losing progress.

THE PROTOTYPE

In 2012, we built a prototype system that could enable an iLVC enabled vehicle to fly adaptive training exercises and record student experiences (as they happened) to a ground based LRS. The goal was to provide an adaptive training exercise that provided real-time reporting of student pilot performance in relation to learning objectives.

In order to build this prototype system we targeted realistic simulation of the actual flight environment. This allowed us to verify the feasibility of integrating this feature into a real aircraft environment. To do this, we integrated our iLVC Pod to a ground based flight simulator. For the research, we utilized a radio simulation to make the communication from the ground and the aircraft realistic. The LRS messages from the aircraft to the ground used a modified automated performance assessment and scenario adaptation software (called the Embedded Instructor) to send out LRS messages at key points during the training. The messages were packed up and sent via the iLVC communication link. On the ground, these new messages were unpacked and forwarded to the LRS for storing. The LRS used was sample code provided by ADL in their Tin-Can prototype section of their website (https://github.com/adlnet/TinCan_Prototypes).

We also built example applications to provide a way for the students and instructors to visualize the results of the training. One example was a web based scoreboard application that utilized the records stored in the LRS to provide a top 10 list. This application ran on the LRS server and periodically queried the LRS for the scores, then sorted and displayed them. A second application ran on an Android tablet as an example of how instructors or administrators could access the data in real time to view the current state of the training as well as the scores from previous attempts.

THE RESULTS

After assembling the system and putting it through initial runs we were able to determine the initial feasibility of running an adaptive training exercise in an iLVC environment. The iLVC hardware was more than adequate to run the software required to do the real-time assessment and the mission adaptation. Also, it was determined that due to the relatively low occurrence of LRS messages, there were negligible impacts to normal iLVC messaging. A notable illustration of robustness was that during demonstrations at I/ITSEC, even though the show floor was saturated with wireless networks, the system worked quite well.

The LRS open-source prototype worked well for these prototypes. The score board application and instructor operator station app were both easy to implement due to the easy access provided by web services of the LRS. Ad-hoc xAPI messages and queries were not lost. Our next experiment attempts to validate that the xAPI and LRS mechanisms work in larger-scale, faster-paced training exercises. We anticipate success in this experiment while we expect that the prototype LRS may need some modifications.

THE POSSIBILITIES

Tracking learning experiences that happen anywhere was the goal of the xAPI. Although we did not have a live aircraft operating in conjunction with the booth demonstrations, we did show that the xAPI could even be used to record those experiences in a standardized way for after-action reviews, for student credit, and for later analysis. While we do not recommend sending *all* simulation or live aircraft data (e.g. every DIS or HLA packet) to the LRS, we do recommend the use of the xAPI to enable relevant training records to be tracked. This would include important milestones or observations in addition to scores and completion records.

Even though the mechanisms of the xAPI are simple, they provide authentication and robustness. Therefore, we are also exploring the xAPI to record on-the-job training and activities that are not specifically training oriented. For example, having the aircraft maintenance Electronic Performance Support System (EPSS) automatically record states for dynamic scheduling as well as maintenance audit logs is a prime candidate for xAPI usage. The experiences could also be as simple as giving a presentation at I/ITSEC. A simple JavaScript call within PowerPoint could be sent when the presenter starts and finishes the presentation. Maybe I/ITSEC booth personnel could get "training credit" for demonstrating technology on the show floor by scanning someone's badge. The I/ITSEC website could give credit to authors anytime someone downloaded their paper. Learning happens everywhere, let's give credit "where" credit is due.

BIBLIOGRAPHY

AICC Frequently Asked Questions. (2013). Retrieved May 30, 2013, from Aviation Industry CBT Committee: http://aicc.org/joomla/dev/index.php?option=com_content&view=article&id=1

Chatham, R. E. (2009). The 20th-Century Revolution in Military Training. In K. A. Ericsson, *Development of Professional Expertise.* Cambridge: Cambridge University Press.

Csíkszentmihályi, M. (2008). *Flow, The Psychology Of Optimal Experience.* Harper Perennial Modern Classics.

e.g. Salas, B. E. (n.d.).

Ericsson, K. A. (2009). *Development of Professional Expertise.* Cambridge University Press.

IEEE Standard for Learning Technology. (2004). ECMA Script Application Programming Interface for Content to Runtime Services Communication.

IEEE Standard for Learning Technology. (2005). Data Model for Content to Learning Management System Communication. *IEEE Std 1484.11.1-2004*.

Schreiber, B. T. (2009). Evaluating Pilot Performance. In K. A. Ericsson, *Development of Professional Expertise.* Cambridge: Cambridge University Press.