An Aerial Refueling Boom Operator ITS Design for Embedded Training

Richard Stottler
Stottler Henke Associates, Inc.
San Mateo, CA
stottler@StottlerHenke.com

Jon P. Deal
Boeing
Oklahoma City, Oklahoma
Jon.P.Deal@Boeing.com

ABSTRACT

This is an investigation into the feasibility and cost/benefit trade-off of an Intelligent Tutoring System (ITS) embedded in an Aerial Refueling Operator (ARO) station in an aerial refueling Tanker. The domain has been investigated, knowledge has been elicited, the design developed, and costs estimated. This instructional and software design and the process used to create it are described in this paper.

The training process designed into the ARO ITS is an adaptation of the current training process described by instructors and documents and observed at Travis Air Force Base. The four main types of ARO skills are: flying the boom, breakaway decisions, checklists, and communications. We looked at on-board and off-board training, initial qualification and refresher training, various types of students with various types of backgrounds, and the full range of Boom Operator tasks, skills, and required knowledge related to the ARO station and aerial refueling. Initial qualification training should follow a building block approach with training broken into a number of training stages which are Introduction and Initial Assessment, Communications Training, Checklist Training, Combined Checklist and Communications Training, Boom Flying Training, and Total Task Training.

The primary goal of the Software Design was to design a set of training systems that implemented the Instructional Design while trying to reduce costs and allow for a system that could be expanded and enhanced in a spiral development methodology. Existing software was reused where cost effective. Components developed for any given trainer are reused in the development of others where possible. The core components of each trainer are: simulated scenario-based evaluation, feedback, and debrief capability. To this could be added a student modeling and instructional planning system.

ABOUT THE AUTHORS

Richard Stottler co-founded Stottler Henke Associates, Inc., an artificial intelligence consulting firm in San Mateo, California, in 1988 and has been the president of the company since then. He has been the principal investigator on a large number of tactical decision-making intelligent tutoring system projects conducted by Stottler Henke including projects for the Navy, Army, Air Force and Marine Corps. Currently he is working on the ARO ITS and a Combined Arms ITS as part of the US Marine Corps Combined Arms Command and Control Training Upgrade System (CACCTUS). He has a Masters degree in Computer Science from Stanford University.

Jon P. Deal is a retired U.S. Air Force Aerial Refueling Operator instructor and evaluator with 17 years and over 3500 hours of experience in KC-135A, KC-135R, and KC-10 aircraft. He currently works as a Customer Training Specialist and Aerial Refueling Operator Subject Matter Expert for Boeing Aerospace Operations in Oklahoma City, Oklahoma on the KC-767A tanker program.

An Aerial Refueling Boom Operator ITS Design for Embedded Training

Richard Stottler
Stottler Henke Associates, Inc.
San Mateo, CA
stottler@StottlerHenke.com

Jon P. Deal
Boeing
Oklahoma City, Oklahoma
Jon.P.Deal@Boeing.com

INTRODUCTION

Many of the clients for a Tanker would like an embedded training capability, some because they will have a small number of the aircraft which doesn't justify the cost of a ground-based simulation trainer and some to avoid the costs associated with traveling from a deployed site to a ground-based trainer. However in an embedded training context (on board, in the air, between missions) an instructor will often not be present. Therefore many of the functions normally provided by an instructor such as student performance, evaluation, coaching, and debriefing will have to be performed by software. This was the main impetus for investigating the feasibility of an Intelligent Tutoring System (ITS) for the ARO. A second benefit from automating instructor functions would be reduced training costs across the entire tanker life-cycle.

This paper first briefly describes the ITS design process and then its results, the instructional design, automatic evaluation mechanisms, and the high level design. The design of one of the specific components, the boom flying trainer is then described. The next step is described in Future Work and the paper is summarized in Conclusions.

ITS DESIGN PROCESS

The main design steps were Initial Investigation, Knowledge Elicitation, Develop Instructional Design, and Develop Software Architecture and High Level Design. Initial investigation involved discussions with aerial refueling instructors to discuss the tasks and decision making required of aerial refueling operators and techniques to instruct them. This included useful scenarios, and methods for evaluating trainee actions in simulated scenarios, determining mastery of required skills and knowledge, remediating deficiencies and selecting appropriate scenarios. This investigation involved going over diverse scenarios faced by aerial refueling operators and their instructors and detailing the decision processes of both. During this investigation, a breadth of diverse scenarios was emphasized over depth, though a few were investigated in depth, in order to develop a good design and cost estimate. This included determining which existing ITS tools and shells might be applicable.

We had several phone conversations with boom instructors which were extremely helpful and resulted in several training documents being sent to us which we used as a basis for these conversations. Much of what we learned was confirmed further at Travis AFB.

We visited Travis Air Force Base, observed a refresher BOT training session and debrief, and spoke to several instructors about initial and refresher training, common errors, specific situations, how they evaluate students, and instructional and debrief techniques. We also discussed the various types of students and the problems and remediations needed for each.

Based on the knowledge engineering, we began the instructional design. We looked at on-board and off-board training, initial qualification and refresher training, various types of students with various types of backgrounds, and the full range of Boom Operator tasks, skills, and required knowledge related to the ARO station and aerial refueling.

We then completed the Software Architecture and High Level Design based on the approved instructional design and existing ITS development tool capabilities. The documentation of the design included a top level hardware architecture showing the relationship between the various individual trainers and a central server, corresponding high-level software architecture which also describes the reuse of components between trainers, and designs for each of the 5 individual trainers.

INSTRUCTIONAL DESIGN

Initial qualification training should follow a building block approach with training broken into a number of training stages which are Introduction and Initial Assessment, Communications Training, Checklist Training, Combined Checklist and Communications Training, Boom Flying Training, and Total Task Training.

The training process is an adaptation of the process described in the syllabus, described by instructors, and observed at Travis Air Force Base. Future boom operator tasks are expected to be very similar to previous boom operators' tasks with the greater complexity allowed for by a computer software interface. The current training process evolved over a large number of years and appears to do a good job, but is very instructor intensive. The current training process is very incremental (each step in the training process is a very small increment over the previous step), which, given the amount of material and level of initial qualification students, is appropriate. Lack of an existing ITS and associated simulation systems prevents the current process from being more incremental and requires intensive manpower.

The four main types of skills are:

- Hand eye coordination: Flying the boom
- Judgment: Breakaway decision: Monitoring and deciding safe versus unsafe conditions
- Checklist Knowledge: Knowing when and which and doing them properly
- Communications: Knowing what to say, when and to whom

At times the boom operator is doing all 4 types of things simultaneously. This requires that each skill be trained to automaticity.

Initial qualification training should, as mentioned above, follow a building block approach with training broken into the training stages. Breakaway Judgment Training is covered during Boom Flying Training and in the Total Task Training stage.

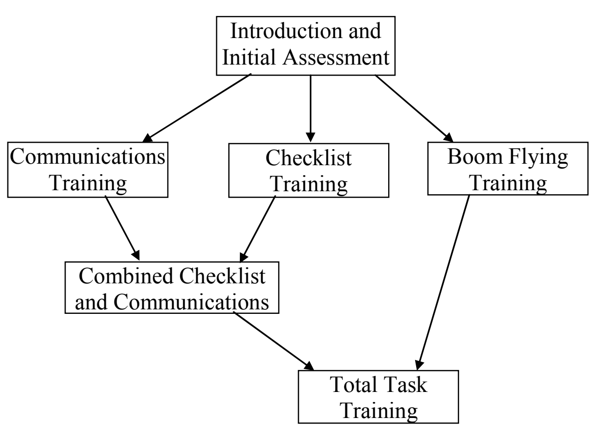

Figure 1. Training Stage Precedence

Introduction and Initial Assessment is done before any other stage. Communications Training and Checklist Training are independent of each other but must both be done before Combined Checklist and Communications Training. Boom Flying Training is independent of all other stages except it must be accomplished before Total Task Training. Total Task Training also cannot be performed until after Combined Checklist and Communications Training is completed. This ordering is shown in Figure 1. Note that this is a partial ordering – there are multiple correct sequences. Additional constraints to the order can be added, if instructors feel it is appropriate to do so. For example, Communications could be trained and therefore assumed to already be mastered in the Checklist Training stage. It should be noted that the preliminary stages serve a dual purpose – acting as both initial training and remedial training. Also, as discussed below, each stage could be divided into smaller parts, such as by sets of or individual checklists. An entire stage would not need to be completed before moving partly forward to the next stage; just training on the relevant checklists.

Boom Flying Training

In the majority of this training, the student practices flying the boom around with specific directions (e.g. "fly a figure 8", "fly a 5 foot box", etc.) and specific coaching and feedback ("more to the left", "you're overshooting", "you're lagging", "don't hold the stick so tight", "try putting your index finger on top"). These directions include telescoping and flying to the limits. The student practices making contacts and other boom flying tasks in simulated scenarios with coaching and feedback. Initial scenarios are easy (no turbulence, good pilots, etc.) and become successively more difficult. Another aspect to be trained in this stage is the perceptual skill of knowing the locations of various aircraft by how they look in an HMD or on a computer display unit. The student is shown various aircraft at various positions and tested on his ability to approximately judge their locations. The student is also tasked with making breakaway judgments. Some of the scenarios involve erratic flying by either or both pilots and or a large degree of turbulence. When the condition is unsafe, the student should hit the breakaway switch. The student graduates from this stage by showing good judgment in the most difficult scenarios. Judging the conditions to be unsafe is complex, involving a number of factors.

Communications Training

The majority of this training is scenario-based practice running through the communications parts of the checklists and other required communications such as calling breakaways and directing the receiver (e.g. "25, down 4"). Communications scenarios only require the student to say the correct words to the correct person.

No other actions are required of him. Other actions required in the scenario occur automatically. A first step would allow the student to read the communications word for word (such as from an amplified checklist), when prompted to in the scenario. After consistent success, the student would be expected to communicate correctly without having the specific words to read. He would be prompted with whatever wording is in the short, operational checklist. If the student has not yet been through the boom flying trainer (where aircraft position perception is taught) then receiver directions would be prompted by obvious indicators (including a display with tick marks indicating various receiver positions). When the student consistently uses the correct syntax, wording, and selects the correct recipient, he "graduates" from this stage.

Checklist Training

The amount of knowledge required to correctly perform all of the checklists is very large. As such, checklists need to be revisited a number of times, in a number of different ways. This requires students to go back through this knowledge a large number of times, but prevents them from getting bored with it, since it is being done differently. Also, since students have different learning styles, more methods of going through checklists are more likely to hit each student's preferred learning style. Each pass described here corresponds to mechanisms described by instructors as a current boom training practice.

The training system first presents and describes the ARO station as both an overview and the details of each switch and button (including the soft screens and buttons). It tests for location and fact recall and tests for speed of location recall.

For each checklist, it presents and describes the checklist along with the rationale for each step. The short, operational version of the checklist is presented first to give an overview in less detail, followed by the amplified checklist. Then the student executes the amplified checklist in slow time (as opposed to real-time) with direct prompts. Then the student reads the amplified checklist for a simulated fellow student and also checks that the fellow student is doing each step correctly. The fellow student occasionally makes mistakes which the real student must catch and correct. Then the student does the checklist while the simulated fellow student (perfectly) reads the amplified checklist. Finally, the student performs the checklist with only access to the short, operational version.

Checklist items that are to check something require the student to touch that thing with his finger. This is true in each stage. After correctly being able to complete the checklists, he must then perform them quicker and while also performing other tasks. If the student has not yet completed the Communications Training stage, communications are said for him. If the student has not completed the boom flying stage, those actions happen automatically for him, as well.

The student graduates from this stage when he has demonstrated the ability to perform the checklists (minus flying and communicating) with little cognitive load, as evidenced by speed and the ability to do other tasks in parallel. These tasks should be other boom operator tasks. Which tasks he has already trained on should be considered in the selection and number of such tasks. Graduation to latter stages may be more fine-grained than this whole stage and might be performed by one specific checklist or a specific set of checklists. This would allow a student to practice checklists he's learned well in the full or intermediate simulator, without having to wait until he has learned them all.

Combined Task Training

The student practices combined checklist and communications tasks in simulated scenarios. Difficulty/complexity of the scenarios starts out low and is increased. If the student has not completed the boom flying stage, those actions happen automatically for him. The student graduates from this stage when he has demonstrated the ability to perform the checklists (minus flying but including communicating) with a high degree of automaticity, as evidenced by speed and the ability to do many tasks in parallel (high difficulty and complexity). Graduation to the last stage may be more fine-grained than this whole stage and might be by one specific checklist or a specific set of checklists for reasons described above.

Total Task Training corresponds to the current Boom Operator Trainer (BOT) training. An open question is, given the previous slow, incremental build up of training, how slow and incremental this stage of training should be. A conservative, slow, incremental approach would have the highest success percentage at the cost of taking more of the student's and the instructional equipment's time. Potentially, an adaptive approach could be used where the student's performance is closely monitored in scenarios and quick, smooth, highly correct performance leads to faster advancement. Ultimately, there may need to be some tuning of the training program with a small trial group of students of various types. The incremental approach is described here with the understanding that students doing well on all aspects can be accelerated. Conversely, students doing poorly on a particular aspect may be remediated by the methods of a previous stage. For example, students having consistent problems with communications may be tasked with getting more communications practice as described in the Communications Training stage.

Scenarios roughly follow the ones that are used for current ARO training, where the initial scenarios are relatively simple with few or no abnormal or emergency procedures, and easy flying conditions and subsequent scenarios being of relatively greater difficulty. Later scenarios may have less incremental buildup and where done, would only involve checklists and procedures new in that scenario. Later scenarios also revisit, without warning, checklists and procedures covered in earlier scenarios.

In the actual ARO station or very close facsimile, in simulated operational scenarios, the student executes amplified checklists in slow time (as opposed to real-time) with direct prompts. Analogously to current training practice, the student reads the amplified checklist for the simulated fellow student and also checks that the fellow student is doing each step correctly. Then the student does the checklist while the simulated fellow student reads the amplified checklist. Finally, the student performs the checklist with only access to the short, operational checklist.

Later scenarios add more refined breakaway judgments. An example is the degree of relative movement of the receiver and refueling mating mechanism (boom or drogue). Generally it is an error to fail to call a "breakaway" in a dangerous situation, but not necessarily an error to call "breakaway" in a less than dangerous situation, unless it occurs too frequently.

The student graduates from each scenario to the next when he demonstrates adequate performance in it. The student graduates from this stage after having graduated from each scenario, implying that he is ready for actual training flights.

AUTOMATIC EVALUATION MECHANISMS

Flying the Boom

Behavior Transition Networks (BTNs) are similar to Finite State Machines (FSMs). An FSM is simply a network of states with specific transitions between particular pairs of states, where each transition has a from-state and a to-state. An FSM is in exactly one of its states, the current state, at a time. Associated with each state may be software that executes while the FSM is in that state. Associated with each transition is a condition. If that condition is true when the FSM is in the from-state of the transition, then the FSM will transition to the to-state. An FSM will have one initial current state that it starts in when it first becomes active.

FSMs are useful because the transition conditions can reference simulation events and values, and trainee actions. Typically, for automatic training evaluation, a portion of the FSM is used to monitor events and values in the simulation, looking for a specific type of situation. This type of situation places the FSM in a specific state. Then the second portion of the FSM monitors and evaluates the student's relevant reactions (or lack of them) to this type of situation. Typically, it writes messages to the trainee interface and/or to a log file that will be presented as the AAR that describes why the actions were correct or incorrect.

For purposes of evaluation in realistic free-play simulations, traditional FSMs have been found to be too restrictive and they have therefore been generalized into Behavior Transition Networks (BTNs). BTNs are very similar to FSMs in the sense of having states, transitions, transition conditions, and a current state, but BTNs have additional capabilities. For example, BTNs have variables that are automatically bound to the events and other conditions in the transition. These variables are easily passed between states and transitions and even across BTNs. The best way to employ BTNs to monitor real-time mission execution is to have a large number operating in parallel where each looks at the situation and student's actions from the perspective of how they handle specific types of situations or apply specific types of principles. [Stottler 2003] describes BTNs in more detail.

There are a number of common errors that separate BTNs will be developed to spot. These would access, from the simulation, the detailed, dynamic boom trajectory and controls data and the detailed dynamic receiver aircraft's position and velocity. There are several types of common errors. Over controlling can be determined by spotting whether the boom directory overshoots the desired position, which may be changing dynamically. Another common error, usually caused by nervousness and holding the stick too tight, is jerkiness. This can be determined by the smoothness of the control inputs. Lagging, often caused by poor hand-eye coordination, can be determined by excessive latency between the desired boom position and the current boom position. The extension rate can be monitored to make sure it is not too fast. When telescoping starts, the position of the receiver's receptacle can be checked to make sure that it is not too far away, which would cause the boom to exceed its outer telescoping limit before contact can be made. The stick inputs can be checked to make sure that the operator is still following the receiver as he starts to extend the boom and is not freezing the stick.

Breakaway Decision Evaluation

There are a number of ways the correctness of the student's breakaway calls can be evaluated, depending on the circumstances. Determining that the student has called "breakaway" requires relatively simple (given the limited vocabulary and syntax) speech to text software. Flying the boom up away from the receiver can also be checked. Certain scenarios may be designed to create breakaway situations. For example, the scenario may involve the receiver flying erratically enough that a breakaway should be called. Such scenarios can be annotated with the point in the scenario where a breakaway should be called and this can be compared to if and when the student actually calls a breakaway. There are also general criteria that can be elicited from instructors which describe situations where a breakaway should be called. This involves a combination of relative positions and velocities (such as excessive closure rates). A different set of criteria relates to a threshold of aircraft oscillation and oscillations in relative positions. General definitions of "too erratic" are also possible to compute. Such general criteria would be implemented using BTNs that access receiver position and velocity, tanker position and velocity, and boom position and movement.

As described above, failure to call a breakaway, when any of the conditions discussed above hold, is an error requiring immediate feedback. However, calling a breakaway when not really warranted, by itself would not be considered an error. An automatic evaluation system would have to log such unexpected breakaway calls as well as the conditions that were true at the time across all scenarios. This requires the system to know who each student is and therefore a logon process is required. After each new unexpected breakaway call, a determination will be made as to whether they were occurring too frequently. Crossing the first such threshold would elicit relatively minor feedback describing the need to, while being safe, actually make contacts and distribute fuel efficiently. Successive thresholds would result in stronger wording.

Checklist Skill Evaluation

Having software which automatically checks that students are executing each step and following the logic correctly in a checklist is very straight-forward. Currently, software exists to quickly develop such systems. A general capability which always performs such an evaluation will exist but should be considered somewhat of a backup to the other techniques described below. These look for common types of errors, which, when the student makes them, can result in more specific and helpful feedback. One common error is the failure to realize that a specific checklist should now be executed. This is evidenced by the failure to execute the first step in the checklist in a timely fashion. Another common error is to execute the wrong checklist, as evidenced by the string of actions being executed matching a checklist different from the correct one. Instructors often can also indicate for each correct checklist, which erroneous ones are likely to be executed. Sometimes a step gets skipped, as indicated by later steps being executed instead of the correct one. Instructors often know which steps of a specific checklist are likely to be forgotten. Sometimes the student follows the wrong branch of a checklist or jumps to the wrong part. Some specific checklists have specific known common errors, sometimes because of the way they appear in the printed checklist book. Another common mistake is to miss steps that aren't on the checklist but still should be performed. These are items that should always be performed in certain circumstances, such as checking circuit breakers, and therefore aren't on the checklist. These can be considered the same as skipped steps, described above. A final error is starting a checklist when there is no need for one. This occurs because the student believes there is some abnormal or emergency condition that is not actually occurring. Again, this can be identified by, when no steps are expected, seeing the first steps of a checklist. Instructors generally can predict which types of situations will tend to mislead students toward which wrong checklists.

Communications Evaluation

Although the radio communications are not enunciated very clearly by students, the fact that just about every word that the boom operator should say is specified by the situation indicates that speech to text translation should work very well. Given a specific syntax and vocabulary, these systems return with a list of words and phrases that were recognized along with a certainty that this information is correct. The grammar and vocabulary can be set dynamically. Thus, when running one specific checklist, at the step that the boom operator must say a set of specific words, the syntax and vocabulary can be set to exactly these words in the correct order. In a different checklist or at a different point in the same checklist, the syntax and vocabulary can be set to a different set of words. Thus, if the student says the correct words, the system will know it with a very high certainty.

For the cases where the student makes minor mistakes, the radio stream can be duplicated and processed differently, with a looser syntax and vocabulary. For cases where the student makes major errors, such as saying phrases from a different checklist or context, a third duplication can be processed with the vocabulary from any of the checklists, along with some additional common words. Processing in this multi-tier fashion allows correct wording to be recognized with a very high certainty, but in cases where certainty of correct wording is low, the system can examine the results of more general speech to text processing performed on one of the duplications.

This system will also have to monitor that the student has selected the correct switch for the correct recipient of his communication.

ARO ITS HIGH LEVEL DESIGN

The primary goal was to design a set of training systems that implemented the Instructional Design while trying to reduce costs and allow for a system that could be expanded and enhanced in a spiral development methodology. The separate ITSs associated with the individual trainers should be as consistent with each other as possible, since the same students and instructors would interact with them. Existing software should be reused where cost effective. Components developed for any given trainer should be reused in the development of others where possible. The core components of each trainer are: simulated scenario-based evaluation, feedback, and debrief capability. To this could be added a student modeling and instructional planning system (SM/IP). Each is designed separately for each trainer. After the basic capabilities are created, additional customization would be performed. Eventually authoring tools could be added. These latter capabilities are allowed for in the design but are not required.

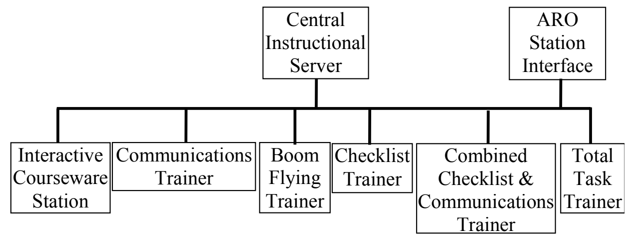

Figure 2. Hardware Architecture

The Communications Trainer includes a simple communications simulator. It is also interfaced to the BTN software, which resides on all of the Trainers. BTNs are used to develop the behaviors to control simulated receiver and tanker pilots, primarily for communications purposes. The Comm versions of the pilots are entirely different than the boom flying versions. Similarly, although also based on BTNs, the Communications Behavior Transition Networks (BTNs) for Evaluation, Feedback, and Hinting/Coaching are entirely different from the boom flying versions.

The Checklist Trainer includes a medium fidelity ARO station simulator, interfaced to the BTN runtime software. SimBionic is used to develop the behaviors to control a simulated fellow student and receiver and tanker pilots. The checklist versions of the pilots are very simple, since they only have to do enough to allow the student to get through the checklists, so they are very different than the Boom Flying and Communications versions. A Task Tutoring Tool (T3), ITS software optimized for procedural training, is used for evaluation, feedback, and hinting. It must also be interfaced to the Simulator, but it should use the same interface as the BTN software.

The Combined Checklist and Communications Trainer primarily combines components from the other two trainers. The Simulator will include the capabilities of both the Communications and Checklist Trainer Simulators. The Tanker and Receiver pilot behaviors will include the behaviors from the Communications and Checklist versions of the pilots. The simulated fellow student is almost identical to the checklist version. The Evaluation and Feedback modules are almost identical to the Communications versions and T3 is configured almost identically to the Checklist version. The difference is that communications are included in the task steps and an interface to the communications evaluator is used to evaluate communications steps.

The Total Task Trainer includes a high fidelity simulator of all aspects of the ARO station. The other components are based almost entirely of components developed for the other trainers. The Tanker and Receiver pilot behaviors will include the behaviors from the Communications and Combined versions of the pilots. The simulated fellow student is almost identical to the Combined version. The Evaluation and Feedback modules combine the Boom Flying and Communications versions. T3 is configured almost identically to the Combined version. Additionally, the Onboard version of the ITS should be very similar to the Total Task Trainer version.

BOOM FLYING TRAINER HIGH LEVEL DESIGN

Figure 4 shows the high level design for the Boom Flying Trainer which includes several components. It is similar to the design of most of the individual trainers. Not shown is an ICW component for initial presentation of material. The Cyan components are required for longer term instructional planning and may not be necessary in an initial version. The initial version would be directed toward real-time evaluation, feedback, and hinting. The yellow components will be implemented using BTNs.

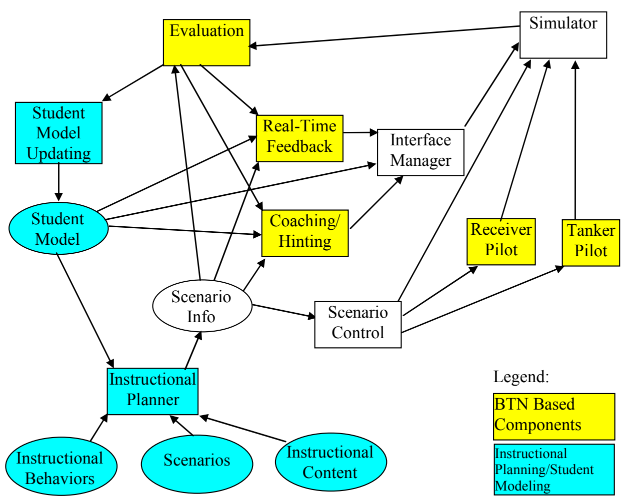

Figure 4. Boom Flying Trainer High Level Design

Simulator and its Interface

The Simulator drives a high fidelity HMD or CDU with a realistic simulation of the tanker aircraft, its boom, and the receiver aircraft. The student interacts with the Simulator through realistic control sticks and a microphone for "breakaway" calls or a breakaway switch. (In the case of the microphone, very simple speech to text software (not shown) to recognize "breakaway" is also present.)

The Simulator Interface provides a means for the Evaluation module to monitor the relevant Simulator variables including receiver translational and angular positions and velocities, tanker positions and velocities, boom positions and velocities, and all student control inputs. The interface will be based on the SISO Draft ITS/Simulation Interoperability Standard (I/SIS) [Stottler et. al., 2005]. It also provides for control of the tanker and receiver aircraft and a mechanism to set turbulence level. It should allow the ITS to start a specific scenario or reset it. It should allow for audio hints and feedback from the Interface Manager and, preferably, the ability to overlay text and graphics in an HMD or CDU. Another preferred capability is to the ability, upon ITS request, to replay specific portions of the trainee's simulated scenario.

Boom Flying Evaluation Module

The Boom Flying Evaluation Module is based on Behavior Transition Networks (BTNs). Separate BTNs evaluate the student's performance along a number of dimensions and send their results to the Real-Time Feedback module for possible immediate action, to the Student Model Updating module, and to the Hinting/Coaching module for possible real-time coaching and hinting. The results are in the form of detected events which correspond to instructionally interesting student actions. Some of these are discrete events and actions and some take place over a period of time. This evaluation module evaluates three broad categories of skills and principles - Boom Flying, Breakaway Decisions, and Receiver Position Perceptions. Each is described in more detail below.

There are several Boom Flying types of evaluations that are examined in parallel and are mostly independent of each other as described in this paragraph. Overcontrolling is detected by calculating overshoot functions and events over time (e.g., hit a desired point with the boom end but with significant residual velocity which forces the student to reverse that velocity). Lag is detected by calculating time difference required to reach desired positions over time and comparing this to an acceptable level. Jerkiness, which indicates an overly tight grip, is calculated as a function of the control stick inputs.

A number of evaluations relate to extension. The extension rate will be examined to make sure that it is not too fast. Also the distance that the receptacle is away from the boom will be checked when telescoping is started, to make sure that it is not too far away. The control stick inputs will be examined when telescoping starts to make sure that the trainee is still following the receiver and hasn't frozen the stick. Also the fact that the boom covers the hole will be checked when telescoping starts.

There are several evaluations that relate to the breakaway decision. Breakaways are indicated by the trainee calling "breakaway" (which is monitored by speech to text software) and/or hitting the breakaway switch. In general, the BTNs examining breakaway decisions monitor (in addition to the microphone and breakaway switch) receiver position and velocity, tanker position and velocity, and boom position and velocity, over time. One BTN may compare the timing of the trainee's breakaway actions to the timing of a specific, predefined event in the scenario. Another, more general one might calculate how erratic the receiver is and compare this variability to different thresholds depending where in the refueling process (at precontact point, moving from precontact to contact point, at contact point, refueling) the receiver is. Variability might be calculated as an amplitude of oscillation, average or maximum error from a desired trajectory, or jerkiness of the path, based on judgments from expert boom operators of simulated receivers and analysis of the associated data. Another might examine closure rates and compare them to different thresholds. One BTN would concern itself with whether the receiver was approaching limits while in contact. A BTN might potentially examine receiver oscillation or relative position oscillation. After a breakaway call, a BTN would ensure that the correct process was being followed, such as making sure that the boom is flown up and away from the receiver. A final behavior would look for excessive breakaways within and across scenarios by keeping track of unexpected breakaways as well as the values of the breakaway indicators discussed above.

Receiver position evaluations are the most straightforward, since they only relate to trainee's judgments of the location of various aircraft in various positions at various points in the refueling process. The trainee is either correct, close, or not close in his judgments.

Real-Time Feedback

The Real-Time Feedback module receives events received from the evaluation module, which mostly consist of mistakes that the trainee is making. (Note that some mistakes are discrete, like the failure to call a breakaway, and others are continuous, like not being able to fly the boom precisely enough.) Based on the level of the student, the priority of the mistakes, and previous feedback, this module chooses the number of events and which ones to provide feedback on. For example, a new student might only receive feedback on one aspect at a time, until it is reasonably mastered; then feedback might shift to the next most important one, etc. An expert student might be able to handle multiple feedbacks at once. An aspect known to be mastered might require no feedback at all, under the assumption that an expert student knows enough to be aware of the problem, or the feedback might be very minimal, just to point it out. This module sends its output to the interface module. This output may be text (intended for text to speech conversion), graphics, or other media. Essentially the same information may be sent in multiple modalities to give the Interface manager different options.

Coaching/Hinting

The Coaching/Hinting module seeks to provide advice or prevent an error before the trainee makes it. It receives instructional events from the evaluation module. These might indicate for example, that although the student hasn't done anything wrong, the degree of jerkiness indicates that he is holding the stick too tight. Coaching advice might be to suggest that the trainee put his index finger on top of the stick. There may be similar advice relating to overshoot, or lagging.

Hints often relate to something that is about to happen that the system believes the student will fail on. For example, just before a breakaway situation arises, a trainee, who historically performs poorly in this type of situation, might receive a hint in the form of a quick review of the relevant breakaway criteria. Similarly, a trainee who tends to freeze the stick when extending might be given the hint to "remember not to freeze the stick during extension" as the boom gets close to the correct location when extension would begin. Most of the hints will ultimately take the form of text to speech audio, but some may also be in a more subtle form, such as highlighting an important gauge to notice or monitor.

Coaching and hinting help novice students progress more rapidly, but are essentially a crutch that needs to eventually be removed. Therefore, the Coaching and Hinting module must try to minimize then eliminate hinting over time, generally with students in the intermediate stage of training. This module will also provide its output to the interface module, typically in different modalities.

Interface Manager

The Interface Manager manages multiple pieces of information for presentation to the student coming from different modules. It must deconflict and prioritize the information to decide what to present and in what order, based on the student model and recent dialog history. For example, novice students may not be able to handle hints and feedback at the same time, or at least not about different topics. Similarly, if the recent dialog from the ITS to the student has concentrated on one aspect, the Interface Manager may choose to filter out feedback and coaching about a different subject. It will pick the modality to present the information and will include text to speech software (for the case that that is the selected modality). More expert students who can handle multiple types of information at once may get it simultaneously using different modalities, for example.

Receiver and Tanker Pilots

These simulated pilots control the aircraft as required for the scenario, the student, and the instructional objectives. For example, beginner students would practice making contacts with very smooth flying aircraft. More experienced trainees would be forced to make contacts on more erratic aircraft. Additionally, an instructional goal might be to check the student's breakaway judgments, which might require the receiver to fly in an unsafe manner. The pilots will be implemented as SimBionic behaviors interfaced to the Simulator.

Scenario Information and Control

Associated with each scenario are instructional goals and events. These require that the scenario controller make certain things happen in the simulated scenario. To ensure this the Scenario Control module can start a specific scenario or reset it (to let a student redo a situation he just failed). It can select the types of receiver aircraft. It can set the turbulence level and other aspects of weather. It also gives high level commands to the simulated pilots.

Debriefer

Not shown in the figure is a Debriefer that receives the instructional events, such as errors or near mistakes such as sloppy flying, from the Evaluation module. After the simulated scenario finishes, the Debriefer goes over the trainee's performance with him. It goes over his mistakes, weak points, strong points and successes. It reviews with him relevant information that appear necessary to help remediate his deficiencies.

Student Model and Updating

Student Modeling should allow a hierarchical representation of tasks broken down into subtasks, skills, and principles. Based on this breakdown, it can automatically create a Bayesian Belief Network which calculates the mastery of each task, subtask, skill, and principle based on scenario evaluation results, treating those results as evidence of the mastery. It also includes Automaticity and Integration at each level in the hierarchy.

A sample of the Bayesian Belief network might include, for example, "True Control Boom Mastery" which has two subnodes, "Control Boom Automaticity" and "Control Boom". "Control Boom" has 4 subnodes – "Perception", "Anticipation", "True Manipulation Mastery" and "Control Boom Integration". This means that to have true control boom mastery the trainee must have mastered the control boom skill and be able to do it automatically. In order to have mastered the control boom skill, the trainee must have mastered the skills of perception, anticipation, true stick manipulation mastery, and be able to integrate those skills simultaneously. Note that a task may have subnodes of subtasks, skills, or principles and skills may have subnodes of subskills or principles.

The Evaluation component outputs to the Student Model Updating component many dimensions of performance information for each item the student was performing. For example, the student may be requested to fly a figure eight with the boom. The Evaluation component might provide grades on a 0 to 1 scale for Overcontrolling, Lag, and Jerkiness for this task. Another example might be grades for making contact with an F-16. The grades from the evaluation module might be horizontal overcontrol, vertical overcontrol, horizontal lag, vertical lag, jerkiness, not freezing, extension rate, covering the hole, and boom in range when extension begins. Some of these map very straight forwardly to student model attributes (e.g. "not freezing" -> "Don't freeze the stick when extending") and some are less obvious and more indirect ("jerkiness" indicates "Grip" is too tight). Thus, the Student Model Updating component references a declarative mapping between performance evaluation outputs and student model attributes. This mapping can also be used by the Coaching/Hinting and Real-Time Feedback components. For example, the coaching mechanism, when it receives the evaluation of excessive jerkiness, could use this mapping to see that this is evidence of too tight a grip. It could offer this information to the student along with the associated suggestion to try putting the index finger on top of the stick.

Instructional Planner, Instructional Behaviors, Scenarios, and Instructional Content

The Instructional Planner is based on BTNs. Instructional behaviors access the student model to determine the next appropriate instructional event. These behaviors control "graduation" from one section of the ITS to another, based on the degree of mastery attained for certain tasks (and/or skills and principles). Prior to graduation, the next instructional event is often scenario practice in a particular scenario which is made up of scenario fragments and a specification of the degree of instructional help (positive and negative feedback, coaching, hinting) that should be provided to the student. The instructional planner has a group of scenarios and scenario fragments that it can use and assemble. Scenario fragments roughly correspond to specific checklists or tasks (e.g., the Precontact or "Boom System Fail Light On" checklists or making contact with a C-130). Scenarios roughly correspond to the overall context that scenario fragments can be embedded into (e.g., one might be a refueling mission which includes N contacts and another one might be an introductory practice mission for flying the boom around in prescribed ways). The instructional planner also decides if a fragment was performed poorly enough that it should be reset and re-executed.

FUTURE WORK

Plans would be to first implement a prototype of the Boom Flying Trainer interfaced to an existing ARO demonstrator, which includes a boom flying simulation as well as simulation of some of the important checklist aspects. The combined ITS prototype/demonstrator would be presented to existing Tanker customers to get feedback before implementing the operational versions.

CONCLUSIONS

The ARO ITS has proven to be feasible. An instructional design was presented which proposes to teach the four main types of skills separately then together in incremental fashion. A software design that implements the instructional plan was presented including mechanisms for automatic evaluation and the individual components for one of the trainers.

REFERENCES

Stottler, R., B. Spaulding, R. Richards, *Use Cases, Requirements and a Prototype Standard for an ITS/Simulation Interoperability Standard (I/SIS),* SISO 2005 Spring Simulation Interoperability Workshop San Diego, CA, April, 2005.

Stottler, R., *Techniques for Automatic AAR for Tactical Simulation Training*, I/ITSEC 2003. Dec., 2003.